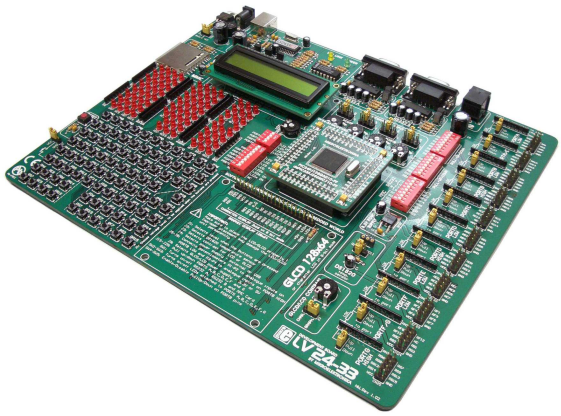
型番:dsPICF-900　　REV.3.80

USB2.0対応 高速プログラマ搭載
PIC24/dsPIC33Fシリーズ開発用統合評価ボード

取扱説明書
お使いになる前にこの説明書をよくお読みの上正しくお使いください。

(C)2007 マイクロテクニカ

ボード本体



製品の概要

PIC24/dsPIC33Fシリーズ開発用統合評価ボード[型番:dsPICF-900、以下dsPICF-900と記載]は、オンボードにUSB接続PIC/dsPICプログラマーを搭載し、最新の16ビットマイコンPIC24シリーズ及びDSP機能内蔵のハイエンドdsPIC、dsPIC33Fシリーズに対応した統合開発ボードです。

ボード上には、100ピンのMCUカード用ソケットが用意されており標準ではハイエンド16ビットPICマイコンPIC24FJ96GA010搭載のMCUカードが付属しています。別途、MCUカードをご購入いただくことでdsPIC33Fシリーズにも対応します。

ボード上に搭載のUSB接続マイコンライターは、最新のPIC24FJ及びPIC24HJシリーズの他、DSP機能内蔵のdsPIC33Fなどのハイエンドデバイスに対応しており、基板上に装着したMCUカードに搭載のマイコンにプログラムを書き込みます。USB2.0対応なのでパソコン側がUSB2.0対応機種であれば高速書き込みができます。また、USB接続を行った場合、USBバスからdsPICF-900の電源を取りますので、別途電源を準備する必要がありません。

ボードに搭載されている各周辺回路に接続されているピンは周辺回路に接続して利用するか又は周辺回路には接続しないで使用するかをディップスイッチやジャンパーソケットで選択できます。また全I/Oピンはボード右側に実装されているヘッダピンから取り出すことができます。また、本セットには6Kバイトまでの HEXファイルが生成できる体験版Cコンパイラーを付属。本書にはチュートリアルも収録されているので、すぐに16ビットマイコンの世界を体験できます。

パッケージの内容

■同梱物
・dsPICF-900ボード本体
・PIC24FJ96GA010搭載MCUボード （ボードに装着済み）
・16文字×2行液晶ディスプレイモジュール （ボードに装着済み）
・USBケーブル
・CD-ROM
・マニュアル(本書)

ドライバのインストール及びパソコンとの接続

■デバイスドライバーのインストール

dsPICF-900をパソコンと接続する前に、デバイスドライバーをパソコンにインストールします。
(Windows98及びMEをご使用のお客様は本項は読み飛ばして、次の「パソコンと接続する」の項よりお読み下さい。)
WindowsXP.Vista,7をご利用のお客様は下記の手順で、ドライバーをインストールします。

1 付属のCD-ROMをパソコンのCD-ROMドライブに挿入してエクスプローラー等で内容を表示します。

2 "USB Programmer Software"のフォルダを開きます。

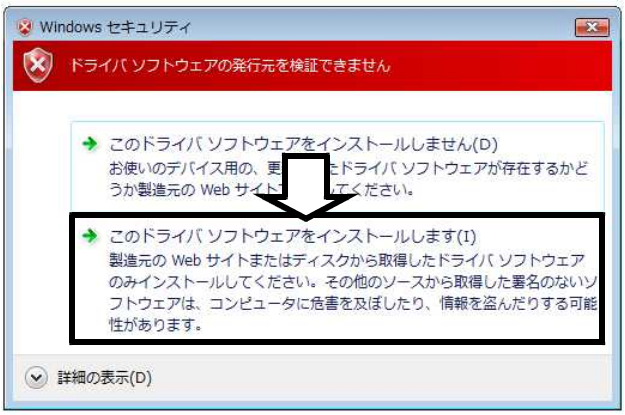
3 "Driver"フォルダを開きます。

4 OSごとにフォルダがあります。ご使用のパソコンのOSにあったフォルダを開きます。
※WindowsVista,7に関しては32ビット版用と64ビット版用がありますので間違えないようご注意ください。

5 開くと実行ファイル、"USB18PRG～.exe"がありますのでダブルクリックして実行します。

6 実行するとダイアログが表示されますので、そのまま"次へ"をクリックして続行します。

7 ドライバーのインストールが開始されます。
WindowsVista環境で下記のような警告ダイアログが表示された場合には、"このドライバソフトウエアをインストールします"をクリックして続行してください。



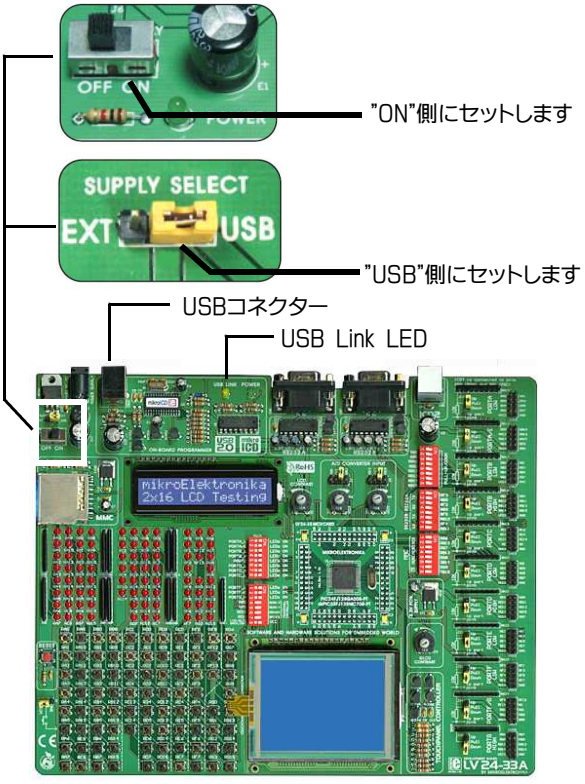
下図のようなメッセージが表示された場合には、"インストール"ボタンを押してください。

8 インストール作業が完了すると、下図のような画面になりますので、"完了"ボタンを押してインストールを完了します。
この時、"状態"のボックスに「使用できます」と表示されていることをご確認下さい。エラーが表示された場合には、ご使用OSの種類と、実行したデバイスドライバーの種類が一致しているかご確認下さい。

■パソコンと接続する

パソコンのUSBポートにdsPICF-900を接続します。

1 dsPICF-900の電源スイッチをONにします。
また、dsPICF-900の電源をUSBのバスパワーより給電しますので、J4の"USB"側にソケットを装着してください。



2 USBケーブルで、パソコンのUSBポートと、dsPICF-900を接続します。

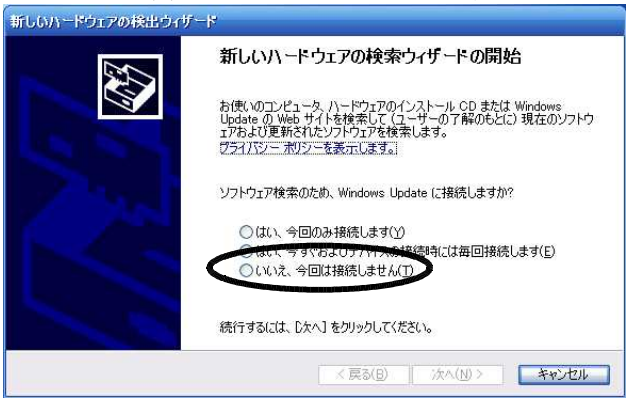
※パソコンのUSBポートがUSB2.0対応の場合には高速書き込みが行えます。USB1.1ポートの場合には通常速度での書き込みとなります。

→接続するとPOWERの緑LEDが点灯します。
→新しいハードウエアの検出ウィザードが自動的に起動します。

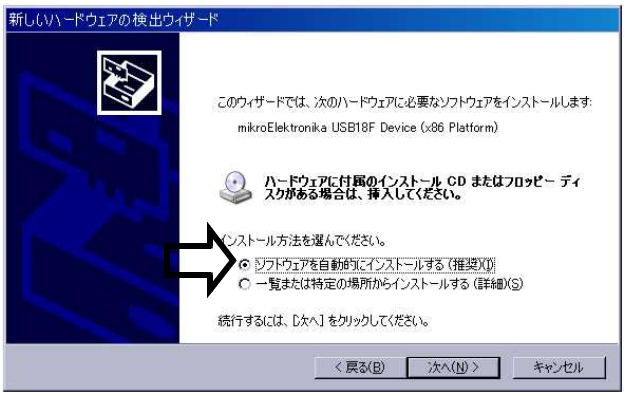
以下OSによって操作方法が異なります。

■Windows2000, XPをご利用の場合

3 下図のようなダイアログが表示されたら"いいえ、今回は接続しません"にチェックを入れて"次へ"ボタンをクリックします。



4 下記のようなダイアログが表示されますので、"ソフトウエアを自動的にインストールする(推奨)"にチェックをいれて、"次へ"ボタンをクリックします。



5 「ソフトウエアをインストールしています。お待ち下さい」というメッセージが表示されます。しばらくするとインストールが完了して下図のような表示となりますので"OK"ボタンをクリックして終了します。

■WindowsVista,7をご利用の場合----------

3 WindowsVista,7の場合には、dsPICF-900を接続すると自動的にデバイスドライバーのセットアップが開始されます。
あらかじめデバイスドライバーは先の手順にてインストールされていますので、WindowsVistaは自動的にデバイスドライバーを検出してインストールを完了します。

接続後しばらく待ちますと、下図のようなポップアップがタスクバーに表示されます。

上図のホップアップが表示されればインストールは完了です。

■Windows98(SE), MEをご利用の場合----------

3 自動的にデバイスが検出されます。
"新しいハードウエアの追加ウィザード"が表示されたら「次へ」を押して続行します。

4 次のダイアログでは,"使用中のデバイスに最適なドライバを検索する"にチェックを入れて,「次へ」をクリックします。

5 "検索場所の指定"にチェックを入れて,「参照」ボタンを押します。
ドライバの場所を指定するダイアログが表示されますのでCD-ROM内の"USB Programmer Software"フォルダ内の"Driver"フォルダにある"WIN98_ME"フォルダを指定してください。CD-ROMドライブがQドライブの場合にはディレクトリは下記の通りとなります。
Q:¥USB Programmer Softeare¥Driver¥WIN98_ME

指定したら「次へ」を押して続行します。
インストールが完了したら,「完了」ボタンを押して終了します。

～ここからは各OS共通の項目です～

dsPICF-900ボード上の"USB Link"の黄LEDが点灯していることを確認してください。
黄LEDの点灯はPC側にdsPICF-900が正しく認識されていることを示します。

※"USB Link"のLEDが点灯していない場合には、正しくドライバーがインストールされていません。再度手順を確認の上、ドライバーをインストールし直しておためしください。

※デバイスマネージャーに"USB Devices by mikroElektronika"が追加されます。

WIndowsVista及び7環境でご使用のお客様へ ユーザーアカウント制御無効設定のお願い

WindowsVista及び7環境で、dsPICF-900をご使用の場合、Windowsに搭載されたユーザーアカウント制御(UAC)機能がUSBポートへのアクセスを拒否することにより、下図のようなエラーメッセージが表示されて、書き込みができないという現象を確認しています。

この現象は、WindowsVista及び7において標準で有効になっているユーザーアカウント制御機能を無効に設定することで回避できます。下記の方法で設定を変更して頂けますようお願い致します。

※なおこのエラーは、WindowsVista及び7以外の環境でお使いのお客様には関係ありません。

■Windows Vistaのユーザーアカウント制御を無効に設定する

1 スタートボタンをクリックして、コントロールパネルを表示します。

2 コントロールパネルから"ユーザーアカウント"をダブルクリックします。"ユーザーアカウント"アイコンがない場合には、ウインドウ左上の"クラシック表示"をクリックして表示を変更します。

3 "ユーザーアカウント制御の有効化または無効化"のリンクをクリックします。

4 "ユーザーアカウント制御(UAC)を使ってコンピューターの保護に役立たせる"のチェックを外します。

5 "OK"ボタンを押して完了します。パソコンを再起動します。

■Windows7のユーザーアカウント制御を無効に設定する

1 スタートボタンをクリックして、コントロールパネルを表示します。

2 コントロールパネルから""ユーザーアカウントと家族のための安全設定"をクリックします。

3 "ユーザーアカウント"をクリックします。

4 "ユーザーアカウント制御設定の変更"をクリックします。

5 画面に表示されるスライドバーを一番下の"通知しない"に設定します。

6 "OK"ボタンを押して完了します。パソコンを再起動します。

ソフトウエアのインストール

開発に必要なソフトウエアをインストールします。なお、ソフトウエアをインストールする際には、必ずUSBポートからdsPICF-900を外した状態で行ってください。

■MPLABのインストール

MPLABはPICマイコンのプログラム開発用の統合開発環境です。PICマイコンの開発元マイクロチップ社から提供されています。MPLABがお手持ちのパソコンにインストールされていない場合には、インストールしてください。

1 CD-ROM内の"MPLAB"フォルダを開きます。
"SETUP.EXE"をダブルクリックして実行します。
"Next>"をクリックして続行します。

2 "License Agreement"のダイアログが表示されたら内容をよくご確認の上、同意する場合には"I accept the terms of the licence agreement"にチェックを入れて、"Next>"クリックして続行します。
なお同意しないとインストールは続行できません。

3 "setup type"の選択ダイアログではすべてのプログラムをインストールしますので"Complete"にチェックを入れて"Next>"をクリックします。

4 MPLABをインストールするディレクトリを指定します。
特に設定する必要がなければデフォルトのままにしておきます。
インストールするディレクトリを変更したい場合には,"Browse"ボタンを押してインストールディレクトリを設定します。
なお変更した場合には,インストールディレクトリをメモするなど忘れないようにしてください。
"Next>"をクリックして続行します。

5 以降のダイアログでは次のように進めます。

■Application Maestro License
→"I accept the terms of the license agreement"にチェックをいれて"Next>"をクリックします。

■MPLAB C32 License
→"I accept the terms of the license agreement"にチェックをいれて"Next>"をクリックします。

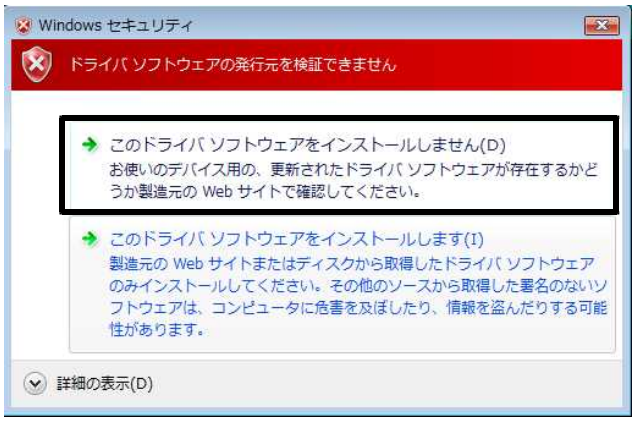
■Start copying files
→"Next>"を押します。
・・・インストールが開始されます。完了までしばらく待ちます。

6 インストールが完了すると、下図のようなダイアログが表示されます。MPLABバージョン8.00以降からはPICCコンパイラーのライトバージョンがMPLABに付属するようになったためです。
ここではOKボタンを押して続行します。

7 下図のようなPICC-Liteのインストール画面が表示されますので、ここでは"Cancel"ボタンを押してインストールをキャンセルします。

※PICC-Liteは、Hi-Tech社のPIC用Cコンパイラーです。MPLAB8.00以降ではライト版(機能限定版)が付属しました。インストールしていただいてもかまいませんが、本キットではこのソフトウエアは使用しません。

8 Windows Vista環境にインストールされている場合には、最後に下図のようなドライバーインストールに関するセキュリティ警告が表示されることがあります。
この場合には、"このドライバソフトウエアをインストールしません(D)"をクリックしてください。
ダイアログが消えたら"Finishボタンを押してインストールを完了します。



■mikroC PRO for dsPIC 2010のインストール

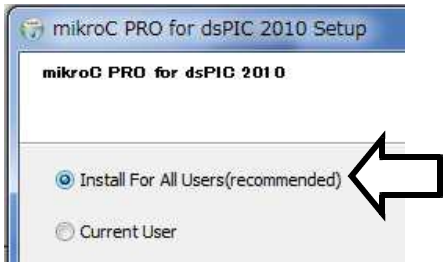
mikroElektronika社からリリースされているPIC24シリーズ及びdsPICシリーズ用ANSI規格準拠のCコンパイラーです。

本dsPICF-900には、生成されるHEXファイルのコードが6KBまでに制限されたCコンパイラーが付属しております。生成されるコードサイズに制限はありますが、使用できる機能及び期限等に制限はありません。この限定版Cコンパイラーをご使用になり、コードサイズの限定がない製品版をご希望の場合には、当方より本dsPICF-900をお買い上げのお客様向けにフル機能版を特別優待価格にて販売しております。

1 CD-ROM内の"Cコンパイラー"フォルダを開きます。
"SETUP.EXE"をダブルクリックして実行します。
"Next>"をクリックして続行します。

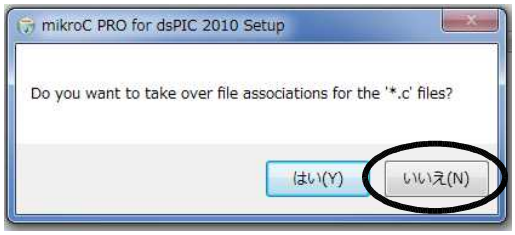
2 "License Agreement"のダイアログが表示されたら内容をよくご確認の上、同意する場合には"I accept the terms in the License Agreement"にチェックを入れて、"Next>"クリックして続行します。
なお同意しないとインストールは続行できません。

3 次の画面では、"Install For All Users(recommended")にチェックを入れて、"Next>"ボタンを押して続行します。



4 "Choose Components"ダイアログが開きますので、"□Examples"にチェックが入っていることを確認して、"Next"ボタンを押します。

5 インストールするディレクトリを指定します。特に変更する理由がなければ、"Install"ボタンをクリックします。インストールが開始されます。インストール開始後下図のようなメッセージが表示されたら、"いいえ"をクリックして続行します。



6 "Completing the mikroC dsPIC Setup Wizard"というダイアログが表示されたら、"Finish"ボタンを押して終了します。
続いて、下記のようなメッセージが表示された場合には、"はい"をクリックします。

7 再度、インストーラーが起動します。画面の指示に従って、インストールを進めてください。"License Agreement"のダイアログが表示されたら内容をよくご確認の上、同意する場合には"I accept the terms in the License Agreement"にチェックを入れて、"Next>"クリックして続行します。

8 インストールを完了すると再度、下図のようなメッセージが表示されます。"はい"を選択して、LvPICFLASH Programmer v1.13をインストールします。

Setup Wizardが表示されますので、画面の指示に従いこちらもインストールしてください。

9 最後に下図のようなデバイスドライバーのインストールメッセージが表示されますが、すでにデバイスドライバーはインストール済みですので、ここでは"いいえ"をクリックしてインストールをせずに完了します。

■PICプログラマ用のソフトウエア

PICへプログラムを書き込む際に使用するプログラマソフトウエアです。mikroCをインストールした際にはLvPICFlashとして書き込みソフトウエアはインストールされていますが、mikroCをインストールしなかった場合には、書き込みソフトウエア単体でのインストールが必要です。
mikroCをインストールした場合には、この手順は飛ばしてください。

1 CD-ROM内の"USB Programmer Software"内にある"SETUP.EXE"をダブルクリックして実行します。インストーラーが起動しますので、"Next"をクリックして続行します。

2 ライセンス許諾画面が表示されますので、同意される場合には、"I accept the terms in the License Agreement"の方にチェックを入れて、"Next"ボタンをクリックします。

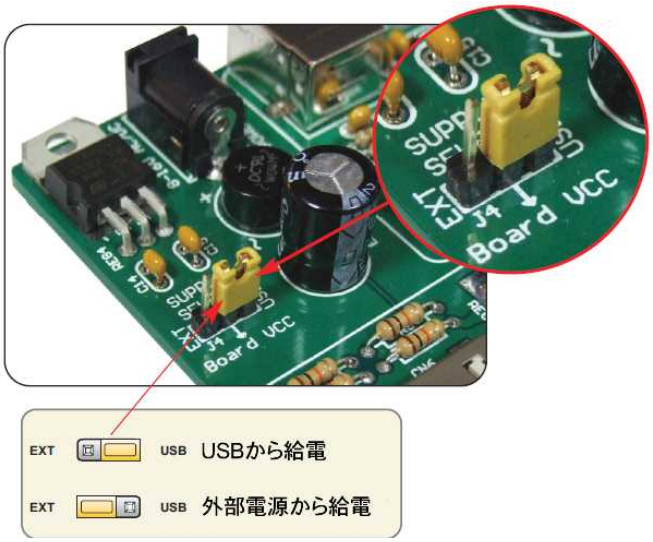
3 次の画面はそのまま"Next"をクリックします。

4 インストールするディレクトリを指定します。mikroCをご使用になる場合には、必ずmikroCをインストールしたドライブと同じドライブにインストールしてください。(例えばmikroCをCドライブにインストールした場合には、こちらのソフトウエアもCドライブにインストールしてください。)
"Install"ボタンを押すとインストールが開始されます。

5 インストールが完了したら"Finish"ボタンを押して完了します。

電源の投入方法

dsPICF-900は、USB接続にてUSBバスから電源の供給を受けることができます。この場合、別途電源を供給する必要はありません。
しかしdsPICF-900側で多くの電流（およそ350mA以上）を消費する予定がある場合にはUSBバスパワー給電では電力が不足する場合があります。この場合には別途外部からACアダプタにて電源を供給する必要があります。給電方法はジャンパーソケットJ4によって設定を行います。なお工場出荷時の設定はUSBバスパワー給電の設定になっています。



■USBバスパワーから給電する場合
dsPICF-900ボード上の"SUPPLY SELECT"ジャンパーピンを「USB」側にショートします。

■ACアダプタから給電する場合
dsPICF-900ボード上の"SUPPLY SELECT"ジャンパーピンを「EXT」側にショートします。

※ACアダプタは、センター極性プラスのDC9Vで500mA以上取り出せるものをご用意ください。ジャック径は2.1φです。（当方にて専用ACアダプタを販売中です）

MCUカードの装着

お買い上げ時は、PIC24FJ96GA010を搭載したMCUカードがすでに装着されています。その他のデバイスを使用する場合には、MCUカードを交換します。交換は下記の手順で慎重に行ってください。

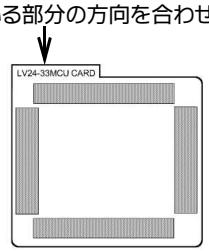
1 dsPICF-900の電源を切断します。

2 現在装着されているMCUカードをゆっくりと上に引き上げます。
取り外しがきつい時は、ピンが曲がらないように注意しながら、MCUカード各側面を少しずつ上方へずらしながら引き上げます。

※取り外したMCUカードは静電気防止用のマットや袋に入れるなどして保管してください。

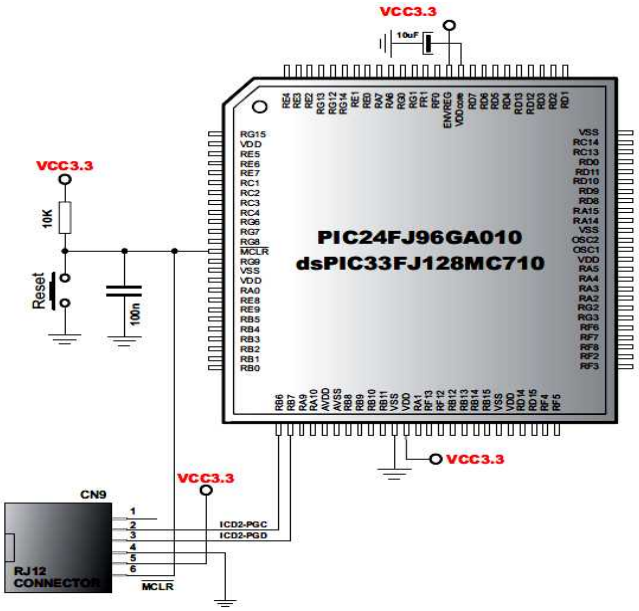
3 MCUカードを装着します。
MCUカードには取り付け方向があります。取り付け方向を間違えるとボードやMCUカードが破損しますので下記のようにdsPICF-900ボード上のシルク印刷に合わせて装着してください。

この突出している部分の方向を合わせて装着してください



MPLAB-ICD2又はICD3を使用する場合

dsPICF-900には、マイクロチップ社純正のICD、MPLAB-ICD2又は後継のICD3を接続するための6ピンモジュラーコネクターが付いています。
ここにMPLAB-ICD2(3)を接続することでdsPICF-900をターゲットボードとして使用することができます。
内部の結線は下記のようになっています。



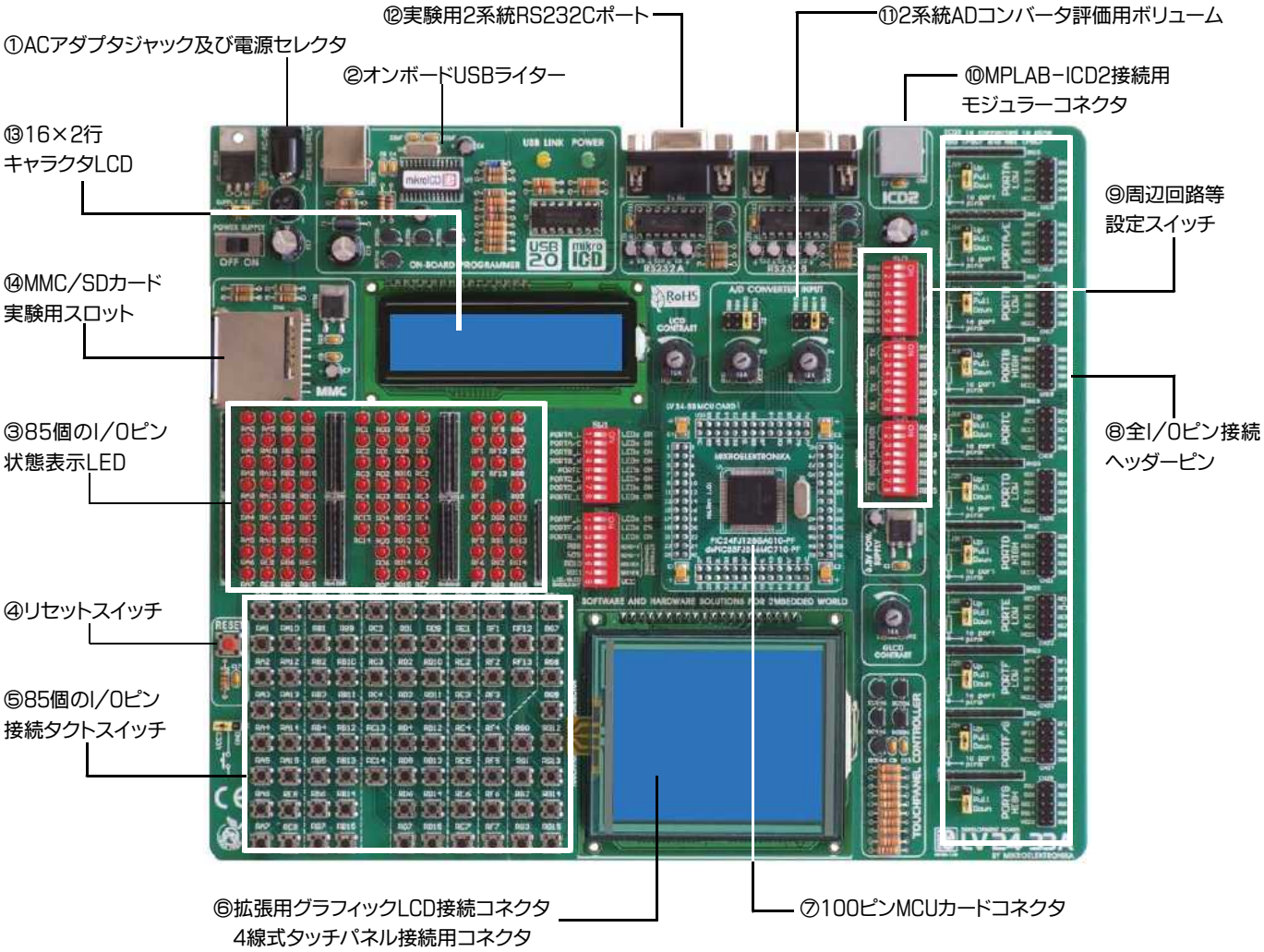
■MPLAB-ICD2使用時のdsPICF-900の電源について
MPLAB-ICD2のVccピンはdsPICF-900のVccピンと接続されています。MPLAB-ICD2接続時には、電源の取り扱い及びMPLAB-ICD2の電源設定を下記のように設定してください。

・dsPICF-900の電源は必ずACアダプタにて外部電源から給電する
・dsPICF-900のUSBはパソコンと接続しない
・MPLAB-ICD2は、"Target has own power supply"に設定する

以上の設定を正しく行ってご使用ください。

■デバッグポートについて
MPLAB-ICD2(3)を用いて、デバッグを行う場合には、コンフィギュレーションビットの設定において、"ICD Communication channel"の設定を、適宜設定してご使用下さい。特に、dsPIC33FJ128GP710拡張アダプタのをご使用の場合には、"ICD uses PGC3/PGD3 and EMUC3/EMUD3"に設定してご使用下さい。

dsPICF-900の各部の説明



---

①ACアダプタジャック及び電源セレクトジャンパ

dsPICF−900の電源をACアダプタから給電する場合には、このジャックにDC9V〜12V出力で2.1φのACアダプタを接続します。センター極性は問いません。USBバスパワー給電では、およそ350mAまで使用できますがそれ以上の電流を消費するような場合にはACアダプタを使用してください。

給電方法はJ4の"SUPPLY SELECT"ジャンパーソケットで選択します。USBバスパワーから給電する場合はUSBと印刷された方に、ACアダプタから給電する場合はEXTと印刷された方にジャンパーソケットをセットします。

②オンボードUSBライター、USBコネクタ

オンボードUSB2.0マイコンライターのUSBポートです。

③85個I/Oピン状態表示LED群

PORTA(RA)〜PORTG(RG)の各ビットの状態を表示できるLED群です。RA・RB・RDは16ビット全ビットにLEDが接続されています。

RC・RE・RF・RGは、ユーザーが使用できるビットに対してLEDが接続されています。

該当のピンがHレベルになると点灯します。

LEDとマイコンのピンを接続するか、接続しないかの設定をディップスイッチSW1及びSW2にて設定できます。

スイッチの"LED's ON"と基板上に印刷された方にスイッチを設定すると、そのポートがLEDと接続されます。基本的に16ビットある各ポートのピンに対して上位8ビットと、下位8ビットの8ビット毎に分けて設定ができます。

なお、一部スイッチの設定が他のポートと兼用になっているものがありますので、基板上のシルク印刷を参考に設定をしてください。

マイコンのI/OピンにLEDが物理的に接続されていると、LED側に電流が流れ電圧降下が発生します。よって、ピンをI/Oピンとして使用する場合、LEDと接続されていると電圧降下により、Hレベル時にVcc電圧となりません。H−LにてVp−pをVcc−GND電圧で使用したい場合にはSW1、SW2の該当ポートのスイッチをOFF側に設定して、物理的にLEDからI/Oピンを切り離してご利用ください。

④リセットスイッチ

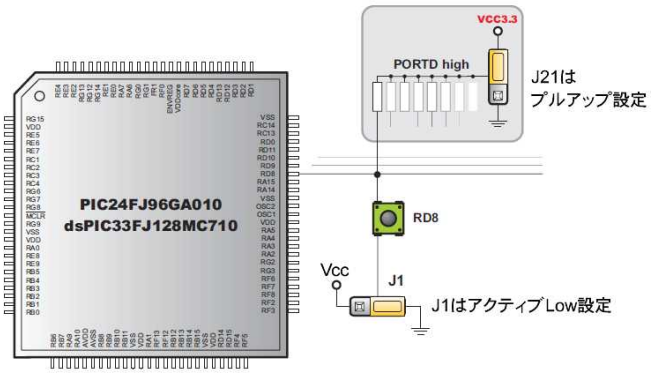
マイコンの$\overline{\text{MCLR}}$ピンと接続されており、スイッチを押すとPICマイコンにハードウエアリセットがかかります。

⑤85個I/Oピン接続タクトスイッチ群

J1のジャンパー設定によって、アクティブHigh又はアクティブLowが選択可能な85個のタクトスイッチです。
タクトスイッチ群の左側にあるJP1をVcc3と刻印された側に設定すると、アクティブHighに、GNDと刻印された側に設定すると、アクティブLowになります。
※アクティブHighは、ボタンを押した時に該当ピンがHレベルになり、アクティブLowは、ボタンを押した時に該当ピンがLレベルになる設定のことです。

各スイッチのプルアップ及びプルダウンは、各ポート8ビット単位(上位8ビット、下位8ビット)で設定できます。プルアップ/プルダウンの設定は、"⑧全I/O外部接続用ピン"の部分でジャンパー設定J15～J25によって設定できます。

押しボタンスイッチの設定は、J1のアクティブLow/High設定と、J15～J25のプルアップ/プルダウンの設定を適切に組み合わせる必要があります。J1をアクティブLow設定にした場合には、該当する押しボタンスイッチの接続されたポートは、プルアップしておく必要があります。下記に例を示します。



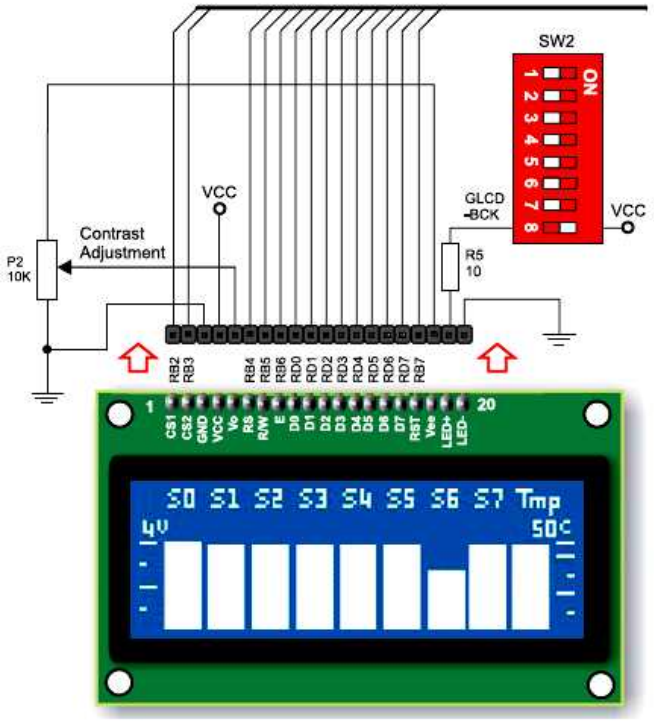
逆に、JP1をアクティブHigh設定にした場合には定常時がLowレベルになっている必要がありますので、JP15～JP25は、プルダウン側にセットすることになります。

なお、PORTBの上位ビットだけは、ディップスイッチSW3によって、1ビット単位で個別にJP18で設定したプルアップ、プルダウンの設定を有効/無効どちらに設定するか決めることができます。
これは、PORTBにはADコンバーターのアナログ電圧入力機能がアサインされているためで、dsPICF-900のボード上⑪のADコンバータ評価用ボリュームを使用する関係上、ビット単位で設定ができるよう設計されています。ADコンバーターを使用する場合には、使用するピンのスイッチをOFF側にセットします。

⑥グラフィックLCD用ピン4線式タッチパネル接続用コネクタ

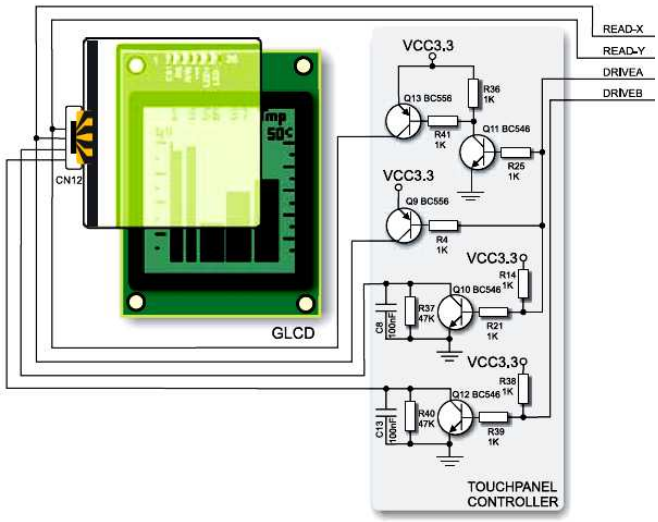
128×64ドットのグラフィックLCDを接続するためのソケットです。また、キャラクタLCDを8ビットモードで使用する場合にもこのピンを使用します。
4線式抵抗膜方式のタッチパネル付きグラフィックLCDも装着できるよう、専用のコネクタも実装されています。

装着したGLCDのコントラストは、P2のボリュームで調整します。内部結線は下記の通りです。



GLCDのバックライトは、ディップスイッチSW2の8番スイッチでON/OFFを行うことができます。

タッチパネル用ソケットは、4線式抵抗膜方式専用で、当方からオプション品としてタッチパネル付きGLCDを販売しております。
タッチパネル用ソケットはディップスイッチSW2のスイッチ4～7を介して本ボードのタッチパネルコントローラー回路部と結線されており、マイコン側からタッチ位置の検出をプログラムから行うことができます。結線は下図の通りです。



⑦100ピンMCUカードコネクタ

100ピンのMCUカードが装着できます。
装着時には、MCUカードの突起部分と、dsPICF-900ボード上のシルク印刷の突起部分の方向が一致するように挿入してください。挿入方向を誤ると、機器を破損しますので十分ご注意ください。
なお、MCUカードの脱着時は必ずdsPICF-900の電源を切断した状態で行ってください。

⑧全I/Oピン接続ヘッダピン群

MCUカードに搭載のマイコンの全I/Oピンが取り出せるヘッダピンです。ポート単位で8ビットずつ取り出せるようになっています。
各ヘッダピンの左側には、ポート単位で8ビットずつプルアップ及びプルダウンが設定できるジャンパーJ15～J25があります。

PORTBは、ADコンバータ電圧入力用のボリュームが接続されている関係上、ディップスイッチSW3にて、上位8ビット(RB15～RB8)は1ビット単位でJ18の設定が有効か無効かに設定できるようになっています。SW3をOFFにすると、そのピンはオープンとなります。

このヘッダピン群に外部回路を接続する場合で、各ピンをデジタルI/Oとして使用したい場合には、使用するポートのLEDを物理的に切り離す必要があります。SW1及びSW2の設定により、使用するポートのLEDを切り離してご使用ください。

⑨周辺回路等設定スイッチ

SW3のディップスイッチは、PORTBの上位8ビット(RB15～RB8)に対して、J18で設定したプルアップ/プルダウンの結線を有効にするか、無効にするかを設定します。ADコンバーターを使用する場合には、アナログ入力となるピンのスイッチをOFFにして使用します。

SW4及びSW5は、dsPICF-900に搭載の各種周辺回路と、マイコンのI/Oピンを物理的に接続するか、しないかを設定するスイッチです。SW4はRS232C関係の配線を、SW5は⑭のMMC/SDカード実験用スロットとの配線をそれぞれ設定するスイッチです。
周辺回路を使用しないとは、電流が流れたり周辺回路からの影響を避けるためスイッチをOFFにして使用します。

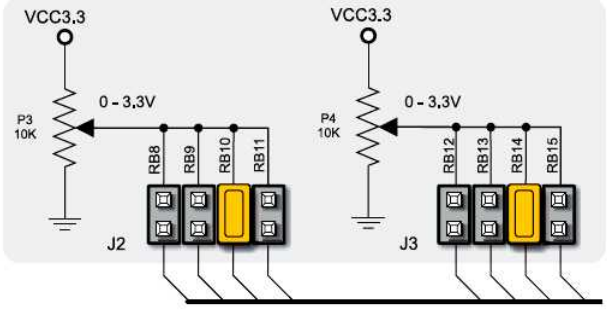
⑩MPLAB-ICD2用モジュラージャック

マイクロチップ社純正のICD、MPLAB-ICD2を接続できる6ピンのモジュラージャックです。MPLAB-ICD2をお持ちの場合、ここにICD2を接続すると、dsPICF-900をターゲットボードとして使用することができます。
詳しくは、本書4ページの「MPLAB-ICD2を使用する場合」の項をお読みください。

※MPLAB-ICD3は本ボードでは仕様上ご使用頂けません。

⑪ADコンバーター評価用ボリューム

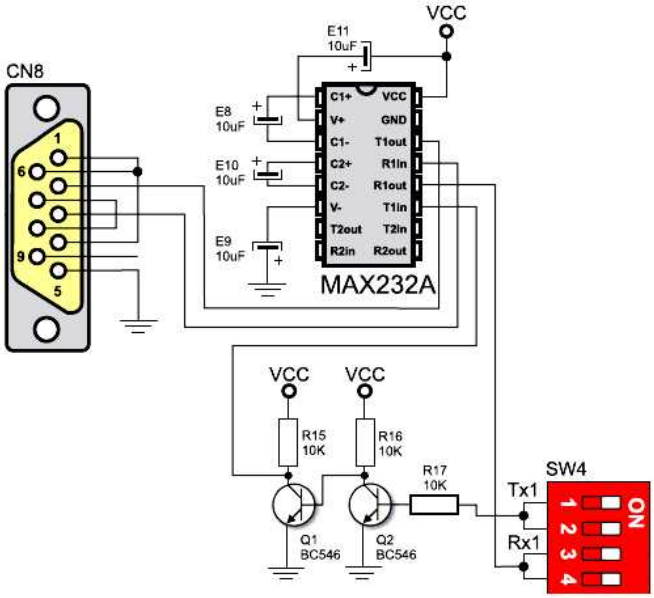
ADコンバータ評価用のボリュームが2系統搭載されています。
J2及びJ3でボリュームを接続するI/Oピンを設定します。



ボリュームの電圧は、0V～3.3Vまでの範囲で変化します。
ADコンバータを使用する場合には、必ずADコンバーターに使用するピンのプルアップ・プルダウンの設定はOFFにしなければなりません。ディップスイッチSW3にて、ADコンバータとして使用するピンの設定をOFFにしてご利用ください。

⑫2系統RS232Cポート

レベル変換IC搭載のRS232Cポートを2系統搭載しています。
左側のコネクタ(CN8)は、"A系統"です。
右側のコネクタ(CN7)は、"B系統"です。
A系統・B系統それぞれのTXピン、RXピンはディップスイッチSW4によって物理的にマイコンのI/Oピンと接続するか、切り離すかを選択できます。スイッチをON側にセットすると、配線が接続されます。
両系統のRX線及びTX線はそれぞれ2つのピンから選べますので、ご使用になるデバイスの仕様やプログラムの内容に合わせて適切に設定してご使用ください。



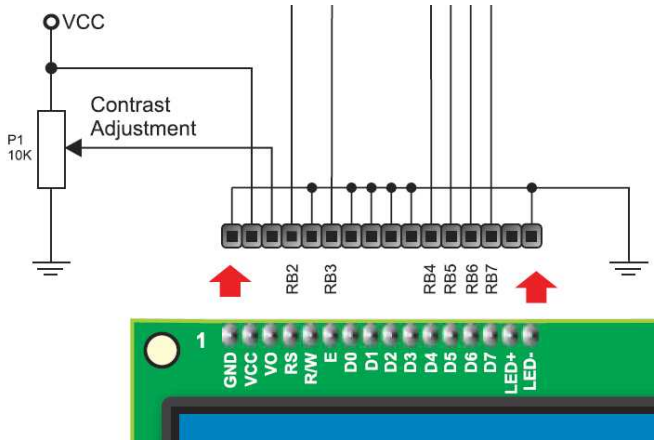
A系統の配線図を上図に示します。
なおCTS線、RTS線は短絡されています。

⑭16文字×2行液晶ディスプレイ(4ビットモード)

16文字×2行のキャラクタLCDが装着できます。
LCDは、4ビットモードで駆動します。LCDのデータ線は上位4ビットを使用し下記のような配線になっています。

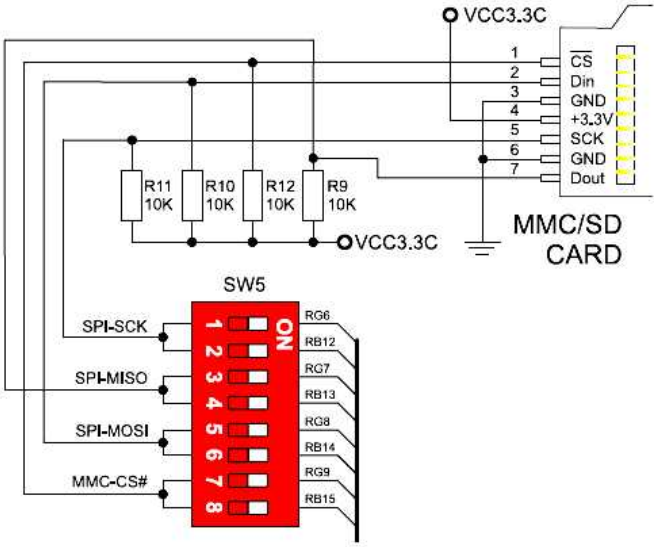
LCD側	マイコン側
RSピン	RB2
R/Wピン	GND(Wモード)
Eピン	RB3
D4ピン	RB4
D5ピン	RB5
D6ピン	RB6
D7ピン	RB7

LCDのコントラストは、P1ボリュームで調節できます。またバックライトはSW2の8番で設定できます。



⑯MMC/SDカードスロット

MMC又はMMCと電気的な互換性のあるSDカードを、SPIモードで制御してアクセスの実験ができます。
メモリーカードのCSピン、SDIピン、SCKピン、SDOピンはそれぞれ10KΩの抵抗でプルアップされ、ディップスイッチSW5を通してマイコンに接続されています。それぞれの信号線をマイコンのどのピンと接続するかをSW5によって設定できます。

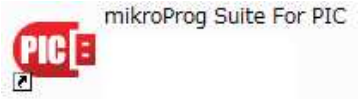


なおメモリーカード用の電源はdsPICF-900内で作っていますので、装着すればそのまま使用できます。

書き込みソフトウエアの使用方法

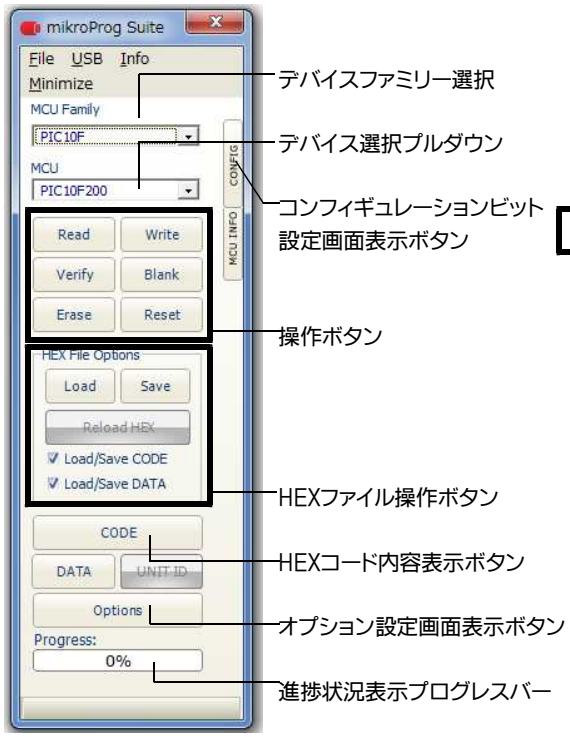
付属のUSB2.0マイコンライターの使い方を紹介します。
マイコンライターは、コンパイラで作成したHEXファイルをPICマイコンへ書き込む際に使用するソフトウエアです。
ソフトウエアは、mikroProg Suite for PICを使用します。

■PICプログラマーの起動
PICプログラマーは、Windowsのスタートメニュー又はデスクトップに作成されたショートカットから起動できます。

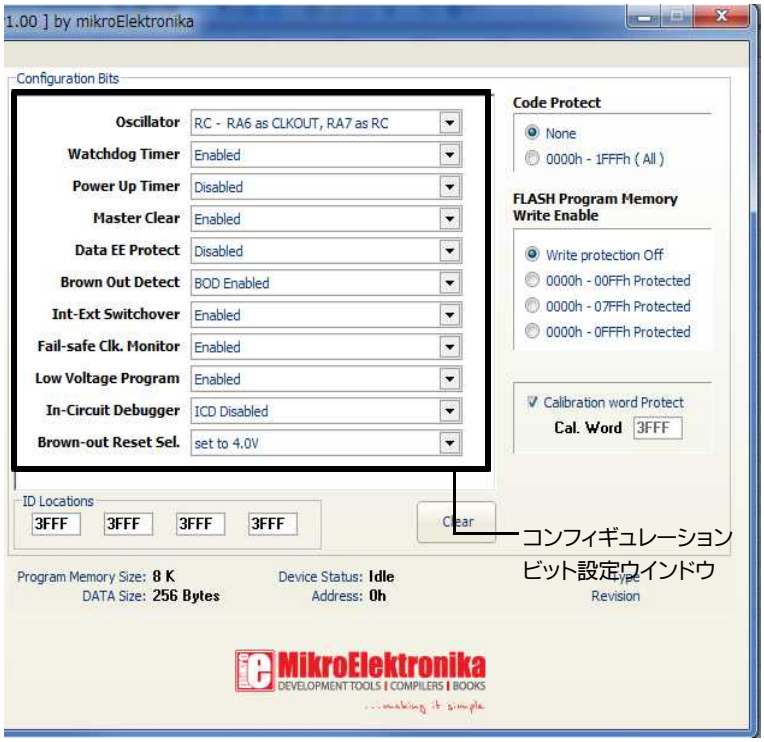


スタートメニューから起動する場合には、"スタート"→"プログラム"→"Mikroelektronika"→"mikroProg Suite For PIC"→"mikroProg Suite For PIC"の順でクリックして起動します。

■アプケーション概要



初期画面



"CONFIG"ボタンを押した時に右側に表示される画面

■操作の順番
次の順番でHEXファイルを書き込みます。

1 デバイス設定プルダウンより、書き込むデバイスを選択します。
最初に"デバイスファミリー選択"プルダウンからデバイスファミリーを選択します。PIC24シリーズ、dsPIC33シリーズが選択できます。

2 続いて"デバイス選択プルダウン"より書き込みを行うデバイスを選びます。

3 書き込むHEXファイルを読み込みます。
HEXファイル操作ボタンの"Load"ボタンを押します。
ダイアログが表示されますので、ファイルを選択します。

→ファイルが読み込まれるとステータスバーに、ファイル名が表示されます。

※同じファイルを再度ロードする場合には、いちいち上記のようにファイルを指定しなくても、"Reload HEX"ボタンを押すことで再読込ができます。

※ファイル名や、ファイルの配置してあるディレクトリ名に日本語や2バイト文字が含まれている場合、正しくHEXファイルの読み込みができません。ファイル名及びディレクトリ名は必ず半角英数字になるようご注意ください。

4 一般的にはこれで書き込み準備は完了ですが、必要に応じてコンフィギュレーションレジスタの設定を確認します。
コンフィギュレーションレジスタは、プログラムを書き込む時にだけしか設定できないPICマイコン全体の動作や基本的な設定などを行う特殊なレジスタです。
コンフィギュレーションビットの設定が間違っているとプログラムや回路が正しくてもプログラムは動作しません。通常、コンフィギュレーションビットの設定内容は、プログラム内に埋め込まれているため、HEXファイルを読み込んだ時点で自動的に設定されます。
しかし、設定が正しくないとプログラムの動作に問題が発生することがありますので、書き込み前に確認されることをお奨めします。

"コンフィギュレーションビット設定画面表示ボタン"をクリックします。右側に設定ウインドウが展開します。

5 必要に応じてコード保護機能も設定します。
コードプロテクションを有効にすると、一度書き込んだHEXデータは読み出せなくなります。コードを不正にコピーされることを防止できます。通常は、"None"及び"Write protection Off"に設定します。

6 設定が完了、確認したら書き込みを行います。
dsPICF-900本体の"USB LINK"の黄LEDが点灯していることを確認してから操作ボタンの"Write"ボタンをクリックします。
書き込みを開始します。
書き込み動作中は、進捗表示プログレスバーに進捗状況が表示されます。完了するまで待ちます。

書き込みが完了すると、プログラムがすぐにボード上で動作を開始します。

■その他の機能

・Read機能
→PICマイコンからデータを読み込みます。
読み込んだデータは、"Code"ボタンで閲覧できます。またメニューバーの"File"→"Save HEX"で保存できます。

・Verify機能
→現在読み込まれているHEXファイルの内容と、PICに書き込まれているプログラムの内容が一致しているか検証します。

・Blankチェック機能
→現在装着されているデバイスのプログラムメモリがブランク(空)かチェックします。

・Erase機能
→現在装着されているデバイスのプログラムメモリの内容を消去します。

主な仕様

電源電圧:	USBバスパワー給電時 DC5V ACアダプタ給電時 DC9V
給電方法:	USBバスパワー又はACアダプタ
USB規格:	Ver,2.0対応
対応OS:	Windows98/ME/2000/XP/VISTA/7
対応デバイス:	PIC24シリーズ及びdsPIC33Fシリーズで MCUカード形式になっているデバイス
生産国:	セルビア/中国/日本

サポート情報

dsPICF-900のサポートを行っております。以下のいずれかの方法でご質問をお寄せください。

■FAX番号　03-3700-3548
■電子メール　support@microtechnica.net

なお、他社製品に関することや自作回路に関するご質問にはお答え致しかねますのであらかじめご了承ください。

ソフトウエアはアップグレードされることがあります。アップグレードされると対応デバイスが増えたりバグが修正されたりします。
アップグレードの情報などは当方のwebページにてお知らせ致しますので、下記webページを定期的にご確認ください。
http://www.microtechnica.net/manual/

PICマイコン用の統合開発環境MPLABはマイクロチップ社製のソフトウエアです。バージョンアップ等の情報についてはマイクロチップ社のホームページより告知されますので下記ページを定期的にご確認ください。
http://www.microchip.com/ （英語サイト）

マイクロテクニカ
〒158-0094 東京都世田谷区玉川1-3-10
TEL: 03-3700-3535 FAX: 03-3700-3548
(C)2007 Microtechnica All rights reserved

PIC24マイコンとdsPIC33Fマイコンの概要

PIC24シリーズは、米マイクロチップテクノロジー社が開発しているPICマイコン初の16ビットマイコンです。

dsPIC33Fシリーズは、16ビットマイコンとしての機能を搭載しながら、デジタル信号処理を行うDSP機能を内蔵しているのが大きな特徴です。dsPICシリーズには、dsPIC30FシリーズとdsPIC33Fシリーズがあり、dsPIC33Fシリーズは最上位に位置する製品です。

PIC24シリーズやdsPICシリーズの詳しい内容については本書ではとても掲載しきれません。専門の書籍やデータシートをご参照ください。本書では、大まかな概要と、dsPICF-900とC言語を使用した簡単なチュートリアルを収録し、いち早く16ビットの世界を体験できるよう内容を編集しました。

■PIC24シリーズの特徴

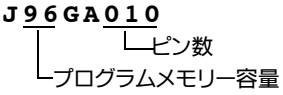
汎用の16ビットマイコンとして開発されたPICマイコンです。外付けの発振子の他に内部に8MHz動作の内蔵発振子を搭載しており、PLLを使用して、最大4倍の32MHzの動作が可能です。dsPICF-900に搭載のMCUカードには8MHzの水晶発振子が実装されています。

PIC24シリーズには、大きく分けて処理速度40MIPSを実現するPIC24Hシリーズと、16MIPSを実現するとPIC24Fシリーズがあります。これら2つの種類は、処理速度の差がありますが、プログラムメモリーの容量やデータメモリーの要領、内蔵モジュールの機能などはほぼ同じです。

16ビットMCUとしての特徴としては、内蔵タイマーが16ビット幅であることや乗算器が17ビット×17ビットと高機能であることなどがあげられます。もちろんメモリーマップドI/Oを採用しているPICマイコンですので、I/Oポートも各ポート16ビット幅となっています。

その他、UARTなどの非同期、同期シリアル通信が各2系統搭載されていることや、内蔵の10ビットA/Dコンバータが1Mspsという高速度で同時に4チャネルをサンプリングできるなど、内蔵モジュールの機能も大幅に向上しています。

PIC24Fシリーズは、型式によってプログラムメモリーの容量と、ピン数が把握できるよう工夫されています。



96はプログラムメモリーの容量で、96Kバイトであることを示しています。010はピン数で、100ピンを示しています。その他64ピンは006、80ピンは008と表します。

なお、PIC24シリーズは電源電圧が2.0V～3.6Vと従来のPIC16FシリーズやPIC18Fシリーズなどの5V電圧と比較して低電圧動作になっています。よってロジックレベルもLVTTLレベルとなっています。

PIC24シリーズの位置づけとしては、8ビットPICの延長タイプであり、8ビットタイプのPICを使用してきたユーザーであってもすぐになじめるように開発されたものです。

■dsPIC33Fシリーズの特徴

dsPIC33Fシリーズは、16ビットマイコンの機能とデジタル信号処理を実現するDSPの機能を内蔵した統合型のコントローラーです。

dsPICシリーズにはdsPIC30Fシリーズもあり、dsPIC33Fシリーズはその上位版という位置づけです。電源電圧も異なり、dsPIC30Fシリーズは+5V動作、dsPIC33Fシリーズは+3.3V動作となっています。

処理速度は、最高で40MIPSと16ビットマイコンではかなりの高速処理を実現しています。

デジタル信号処理では、外部から入力されたアナログ信号を内部で高速にデジタル処理をするため高速なADコンバーターが要求されます。高精度な12ビットのADコンバーターにより最高で200kspsという高速サンプリングができます。また、10ビットのADコンバーターでは最高で1Mspsの高速サンプリングができ、高い周波数のアナログ信号を扱うことが可能となっています。

dsPIC30Fシリーズ並びにdsPIC33Fシリーズは用途に合わせて大きく3つに分かれています。モーター制御用、センサー用、汎用用途と分かれています。それぞれ特徴がありますので使用する際には、データシートをご覧ください。

dsPICシリーズは、PIC24シリーズとはその目的が異なり、8ビット版PICの延長というよりかは、DSP機能を搭載しているということに特徴があります。

～コラム～

大は小を兼ね‥ない?

PICのラインナップは拡充し、現在はCPUコアを32ビットにしたPIC32MXというシリーズも販売されています。今後PICは大型化していく傾向にあるのでしょうか‥?

実は、そうではありません。実際2009年末現在でも最も売り上げの多いPICは8ビットタイプのものです。長年8ビットの代表的なワンチップマイコンとして君臨するPICは、32ビット版が登場した現在でも8ビットタイプが最も選ばれているのです。

ワンチップマイコンの世界では、大は小を兼ねません。常に高機能でピン数が多ければよいか‥といえばそうではありません。アプリケーションの内容によっては大きさは小さい方がいいですし、速度が速くなれば消費電力も大きくなります。マイコンは適材適所、最も最適な品種を選ぶのが賢い方法です。

PICでは機能と大きさを最小限に絞り込んだ6ピンのタイプも(PIC10Fシリーズ)も販売され、需要もあります。16ビット、32ビットタイプが発売されて終息方向に向かうのかと思われたPIC16Fシリーズも機能強化されたタイプが次々とラインナップに加わっています。

C言語とは

C言語は、ANSIという団体によって規格化された高級言語です。高級言語とはアセンブラ言語のように機械語に近いレベルの開発言語ではなく、より人間がわかりやすい言語体系の開発言語を指します。C言語を導入することでより簡単に、合理的にPICマイコンの開発ができるようになります。

C言語では様々なデータ型を扱えるほか、データを指し示すポインタや配列、関数などが使用でき、より高度なプログラムを記述することができます。

本dsPICF-900に付属のmikroCコンパイラーはANSI規格に準拠したdsPICマイコンと、PIC24シリーズなど16ビットタイプのPIC専用のCコンパイラーです。体験版のため、開発できるプログラムのコードサイズが最大6Kバイトに限定されていますが、その他の機能等はすべて製品版と同様に使用できます。期間制限もありません。

mikroCは、一般的なC言語としての機能の他にPICマイコンの機能をフルに使用できるよう様々な組み込み関数を搭載しています。組み込み関数を使用することで、LCD制御やADコンバーターの操作、UARTなど様々な機能を簡単に実現できます。PICマイコンの開発に便利な様々な組み込み関数が搭載されています。

チュートリアル　～dsPICF-900をmikroCで使おう～

早速付属のmikroCを使ってC言語でプログラムを作り、実際にdsPICF-900のボード上で動作させてみましょう。本チュートリアルを一通り試すことで、dsPICF-900の使い方及びmikroCの使い方を体験・習得できます。

なお本マニュアルではC言語の基本から解説することはできませんので、C言語の基本については専門書籍をご参照ください。

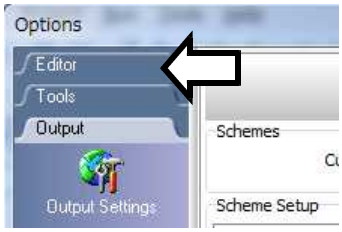
■開発環境の設定

では最初にmikroC Cコンパイラーの開発環境を設定しましょう。

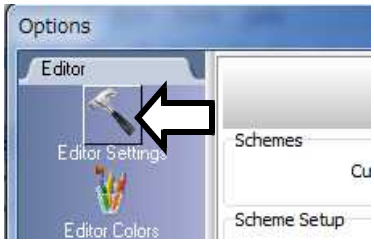
1 mikroCを起動します。
"スタート"→"プログラム"→"Mikroelektronika"→"mikroC PRO for dsPIC"→"mikroC_dsPIC"の順でクリックして起動します。

2 起動すると、開発画面が表示されます。
初期設定では、文字が小さく見にくいのでフォントサイズを大きく変更しましょう。

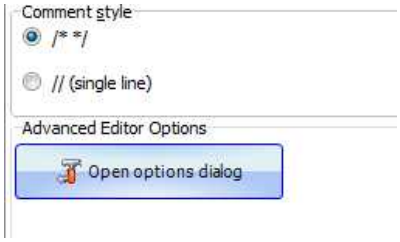
3 メニューバーから"Tools"→"Options"をクリックします。ダイアログが表示されますので、左側のタブから"Editor"をクリックします。



4 続いて、"Editor Settings"のボタンを押します。



5 "Open Option dialog"ボタンをクリックします。



6 "Editor Options"ダイアログが表示されますので、画面右下の"Editor Font"部分にある"Font"ボタンをクリックすると、フォントを選択できるダイアログが表示されますので、サイズを調節してください。

※実用的な大きさとしては、画面の解像度にもよりますが、14ポイント程度が見やすいです。

※右上にある"Font"ボタンではありませんので注意してください。

※フォントの種類は変更しないでください。

設定したら"OK"ボタンを押して完了します。さらにOptionsの画面では、"Apply"ボタンをクリックして設定内容を反映させた後、"OK"ボタンを押して完了します。

※その他にも詳細な設定が可能ですが、トラブルを避けるためフォントの変更以外は行わないことをお奨めします。

チュートリアル①　～PORTBの制御とボタン処理～

では早速プログラムを作って開発を体験してみましょう。最初のチュートリアルでは、まず全体の体験ということで、簡単なプログラムを作ります。

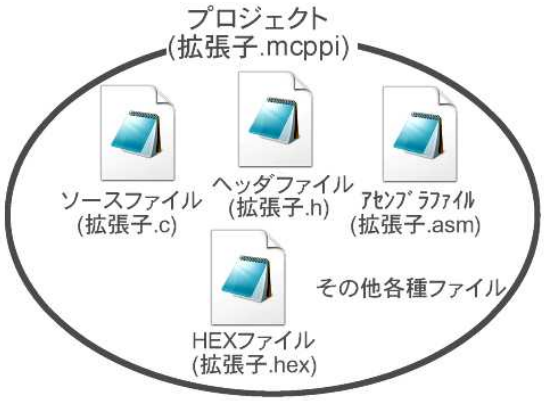
16ビットの変数を用意して、そこに入れた値を2進数でPORTBに出力するプログラムを作ってみましょう。変数を2つ用意して、外部のスイッチ入力によって、どちらの変数の値をRBに出力するのか指定できるようにしてみます。

デバイスは標準で付属しているMCUカードに搭載のPIC24FJ96GA010を使用することにしましょう。

【1】プロジェクトの新規作成

プログラムを作り始める前に、プロジェクトを新規に作成します。mikroCでは1つのファームウエアに対して1つのプロジェクトとして、プロジェクト単位で管理します。

プロジェクトには、これから作成するC言語のソースプログラム(拡張子.c)や、場合によって使用することとなるヘッダーファイル(拡張子.h)などのファイルが登録されます。その他、コンパイル後のHEXファイルなどもプロジェクトに含まれる1ファイルとして管理されます。



よくある間違いとして、プロジェクトの意味を理解していないと画面上に表示されているソースプログラムをコンパイルしているつもりなのに、実際には別のソースプログラムがコンパイルされていて思ったような動作をしなかったり混乱してしまったりというトラブルに繋がります。新しいプログラムを作るときには、必ずプロジェクトを新規に作成するようにしてください。

1 新規にプロジェクトを作成してプログラムを作っていきましょう。
メニューバーから"Project"→"New Project"をクリックします。
mikroCではウィザード形式でプロジェクトを新規に作成します。ウィザード画面が表示されたら、"Next"ボタンを押して続行します。

今回使用する"P24FJ96GA010"を選択して、"Next"ボタンを押します。

2 続いてクロック周波数を設定します。今回は、dsPICF-900に最初から付いている8MHzを使用することにします。"Device Clock"の部分に、"8 と入力して 8MHz に設定します。
小数点以下は自動的に0が挿入されますので、単に 8 とだけ入力してください。

"Next"ボタンを押します。

3 プロジェクトの保存場所と、プロジェクト名を指定します。
ダイアログの右端にあるファイルボタンを押すとディレクトリを指定するダイアログが表示されます。プロジェクトを作成するディレクトリと、プロジェクト名を入力して"保存"ボタンを押します。

ディレクトリ名及びファイル名には、日本語や全角文字などの2バイト文字は使用できません。必ず半角英数字で指定できるディレクトリ及びファイル名としてください。
デスクトップやマイドキュメントなどは日本語が含まれてしまうことがあり、お奨めできません。

ここでは、例としてCドライブの直下に"mikroC_project"というフォルダを作り、そこに"TEST01.mcpds"というプロジェクトを作成することにします。ディレクトリ名は次のようになります。

C:¥mikroC_project¥TEST01.mcpds

※プロジェクトを作成したフォルダには各種ファイルが作られますので、なるべく分かりやすいディレクトリにすることをお奨めします。

"Next"ボタンを押します。

4 続いて "Step4/6"、"Step5/6" のウィザードでは何も変更等はせずに"Next"ボタンを押します。
最後に"Finish"ボタンをクリックして完了します。

【2】プログラムの記述

ではプロジェクトが作られましたので、プログラムを記述しましょう。mikroCでは、新規にプロジェクトを作成すると、"プロジェクト名.c"という拡張子がcのファイルが自動的に作られます。基本的にはこのファイルにC言語のプログラムを記述していくことになります。

新規にcファイルが作られると、main関数だけが記述された状態になります。では早速プログラムを記述してみましょう。

C言語では大文字・小文字は基本的には関係ありません。但し、統一した方が見やすいので、一般的には関数や変数は小文字で、レジスタ名は大文字で記述することが多いです。
プログラムは見やすいようにタブやスペース、改行を適宜入れながら記述していきます。コンパイラは、スペースや改行は無視します。

```
void main(){

  unsigned int a , b ;
  TRISB = 0 ;
  TRISE = 0x3;
  PORTB = 0 ;
  a = 5200 ;
  b = 30000 ;

  while(1){
     if (PORTE == 0x1) PORTB = a ;
     if (PORTE == 0x2) PORTB = b ;
  }
}
```

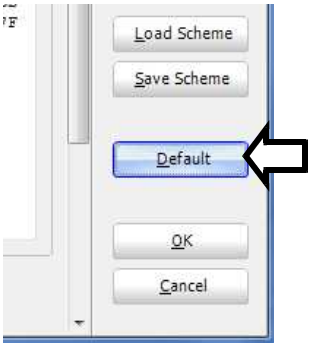
【3】コンフィギュレーションビットの設定

PICには、コンフィギュレーションビットという特殊なレジスタがあります。これは、PICの動作に関わる最も基本的な設定を行うもので、プログラムをPICに書き込む時に設定するものです。
代表的なものには、例えば発振子の種類の指定があります。発振子は大きく分けて外部発振子か、内蔵発振子かに別れ、さらに外部発振子の場合にはその周波数によって、設定が分かれます。その他、ウォッチドッグタイマーのON/OFFの設定などを行うのがコンフィギュレーションビットです。

mikroCでは、コンフィギュレーションビットの設定は、生成するHEXファイルに埋め込んだ形でコンパイルを行います。HEXファイルにコンフィギュレーションビットの設定を埋め込んでおけば、書き込みの時、いちいち設定をしなくても、書き込みソフトウエアはHEXファイルを読み込むと自動的にコンフィギュレーションビットの値をセットするので、手間が省け、間違いも少なくなります。次の手順でコンフィギュレーションビットの値を設定しましょう。

1 メニューバーから"Project"→"Edit Project"を選択します。

2 ダイアログが表示されます。細かな設定は色々とありますが、すべて解説することはできませんので、ここでは必要最低限の設定のみ設定します。最初に最も基本的な設定を読み込みますので、"Default"ボタンをクリックします。



3 プルダウンの一番上"Primary Oscillator Select"の所をクリックして、プルダウンから"HS Oscillator Enabled"を選択します。
その他の設定はそのままでかまいません。

このPrimary Oscillatorとは、PICの発振子の種類を指定するもので、外部発振子の場合、水晶発振子で4MHzの場合にはXTに、5MHz以上又は発振子にセラミック発振子を使用している場合には、HSに設定します。

dsPICF-900のMCUカードには8MHzの水晶発振子が取り付けられていますので、ここでは、HSに設定をしたというわけです。
その他諸処設定はありますが、ここではデフォルト設定を使用します。

※コンフィギュレーションビットの設定については、デバイス毎に異なります。データシートに詳しい記載がありますので、使用するデバイスのデータシートを必ず参照ください。コンフィギュレーションビットの値が間違っていると、プログラムが正しくても動作しなかったり、予期しない動作をしたりすることがあります。

4 設定が完了したら、"OK"ボタンを押して完了します。
なお、以後すべてのチュートリアルで本設定をする必要がありますので、この設定について覚えておいてください。

【4】ボードの設定とコンパイル及び書き込みの実行

C言語で書かれたプログラム(ソースプログラムといいます)を、PICに書き込める機械語のHEXファイルにする作業をコンパイルと呼びます。mikroCとdsPICF-900の組み合わせの場合、コンパイルとコンパイル後に生成されたHEXファイルの書き込みを連動して行うことができます。コンパイルでエラーが発生した場合には、書き込みは実行されません。

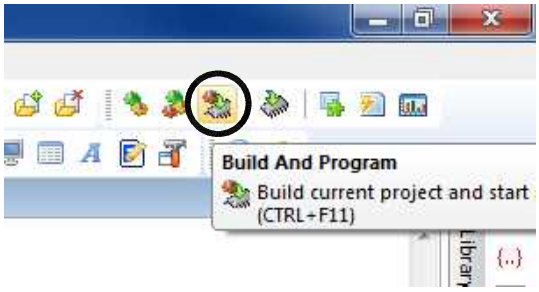
ここでは最初にボードの設定を行った後、コンパイル→書き込みを実行してみましょう。書き込み後はすぐにプログラムが動きますので、書き込みを実行する前に、dsPICF-900のボードの設定をしておく必要があります。

1 dsPICF-900ボードの各種設定を行いましょう。今回はRBはLEDに接続し、REは下位ビットを定常時プルダウンに設定し、タクトスイッチが押された時にVcc(3.3V)に接続されるようアクティブHigh設定にします。次のように設定してください。

・ディップスイッチSW1とSW2
→"PORTB_L"と"PORTB_H"のスイッチをON側にします。
SW2はすべてOFF側にします。
・ジャンパーJ22(PORTE LOW)
→"Pull-Down"間(下側)をジャンパーソケットでショートします。
・ジャンパーJ1(タクトスイッチ設定)
→"VCC3"側をジャンパーソケットでショートします。

上記のように設定すると、PICのPORTBのピンはLEDに接続され、該当のビットがHレベルになるとLEDが点灯します。
また、PORTEは常時Lレベルで、タクトスイッチが押されると、該当ビットがHレベルになります。

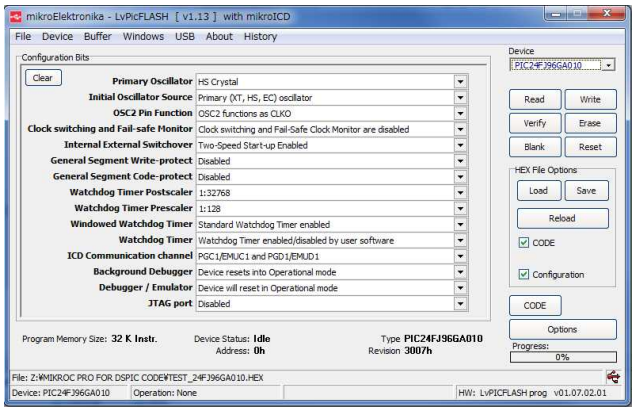
2 設定ができたら、コンパイルと書き込みボタンをクリックします。
ツールバーの"Build and Program"ボタンを1回クリックします。



似たようなボタンがいくつかありますので、間違えないようにしてください。

3 クリックするとコンパイルが始まります。
ウインドウ下部のメッセージウインドウにコンパイル結果が表示されます。エラーがなければ、"Finished successfully"と緑の文字で表示されます。
エラーが表示された場合には、プログラム内容に間違いがないか確認してください。
生成されたHEXファイルは、プロジェクトを保存したフォルダ内に生成されています。

4 コンパイルが正常に完了すると自動的に書き込みソフトウエアが起動し、書き込みを実行します。



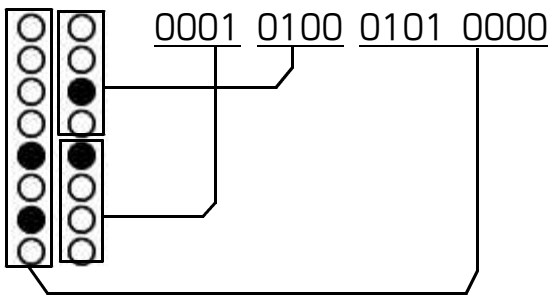
進捗状況がプログレスバーで右下に表示されます。
上記ソフトウエアの"Configuration Bits"と書かれたプルダウンのまとまり部分が先に設定したコンフィギュレーションビットの設定です。

【5】動作確認

書き込みが完了するとプログラムがdsPICF-900ボード上ですぐに動きます。動作を確認してみましょう。

dsPICF-900のタクトスイッチ群のなかにあるRE0のスイッチを押します。RBに接続されたLEDが点灯します。RE0なので変数aの値、ここでは5200が2進数でLEDに現れます。

5200は2進数表記では 0001010001010000 となります。
左側が上位ビット、右側が下位ビットです。RB15が上位ビット、RB0が下位ビットになりますので、下記のように点灯しているはずです。



※○は消灯、●は点灯

同様にして、RE1のスイッチを押すと変数bの値、すなわち30000が2進数としてRBのLEDに表示されます。

【6】プログラムの解説

void main(){…}
main関数です。C言語では1つのプログラム内に必ず1つのmain関数が必要です。voidはmain関数の型で戻り値がない関数であることを宣言しています。{ }で囲まれた範囲がmain関数の範囲となります。

unsigned int a , b ;
符号なしのint型変数を宣言しています。C言語では文の最後には必ず ; (セミコロン)をつけます。
符号なしのint型の場合扱える値は、0～65535までです。

TRISB = 0 ;
TRISE = 0x3;
TRISレジスタは、ピンの入出力方向を設定するレジスタです。
該当のビットが0の場合は出力、1の場合は入力となります。本チュートリアルではRBは全ピン出力、REはRE0とRE1が入力なので0x3として設定しています。
C言語では16進数の値を表現する時は数値の前に0xを付けます。

PORTB = 0 ;
PORTBに0を代入しています。 =(イコール1つ)は代入演算子で右辺の値を左辺に代入します。mikroCでは「レジスタ名=値」の書式でレジスタに値を代入できます。

a = 5200 ;
b = 30000 ;
int型変数に値を代入しています。

while(1){…}
while文は( )内の値が1又は真であれば{ }内の処理を繰り返す繰り返し文です。ここでは評価式を1としていますので永久に{ }内が実行されます。
C言語では処理を永久ループさせたい場合にはよくwhile文を使用します。

if (PORTE == 0x1) PORTB = a ;
if文を使用した入力判定部分です。
()内の評価式が真の時にその後ろに記述された文が実行されます。
mikroCでは、レジスタの値を直接評価式の中で判定することができます。
==(イコール2つ)は比較演算子の「等しい時」を表す演算子です。
よくある間違いとして=を1つしか記述しない場合があります。=が1つの場合には代入演算子となってしまい正しく動作しません。文法上は間違いではないのでエラーがでないため間違いを見逃してしまいプログラムが思うように動かない･･･ということがあります。

PORTEは、スイッチが押されるとそのビットがHになります。RE0のタクトスイッチが押されると、0000 0000 0000 0001 となり、PORTEの値は1となります。同様にして、RE1のタクトスイッチが押されると、0000 0000 0000 0010 となりますので、PORTEの値は2となります。これをif文で判定して処理を分岐しているわけです。

PORTB = aで、変数aの値をそのままPORTBレジスタに代入しています。

チュートリアル② ～LCDの使用と、配列・ポインタ～

■プログラムの作成
dsPICF-900に装着されている2行16文字のLCDに文字列を表示します。C言語では、文字列を扱う場合配列を使用します。
固定した文字列をLCDに表示させてみましょう。
プログラムの作り方は、チュートリアル①と同じです。新規にプロジェクトを作成し、新しいプログラムとして開発します。"Edit Project"からコンフィギュレーションビットの値を設定するのを忘れないようにしてください。

```
sbit LCD_RS at LATB2_bit;
sbit LCD_EN at LATB3_bit;
sbit LCD_D4 at LATB4_bit;
sbit LCD_D5 at LATB5_bit;
sbit LCD_D6 at LATB6_bit;
sbit LCD_D7 at LATB7_bit;

sbit LCD_RS_Direction at TRISB2_bit;
sbit LCD_EN_Direction at TRISB3_bit;
sbit LCD_D4_Direction at TRISB4_bit;
sbit LCD_D5_Direction at TRISB5_bit;
sbit LCD_D6_Direction at TRISB6_bit;
sbit LCD_D7_Direction at TRISB7_bit;

void main(){
     char txt1[] = "HELLO";
     char txt2[] = "WORLD";

     ADPCFG = 0xFFFF;

     Lcd_Init();

     Lcd_Cmd(_LCD_CLEAR);
     Lcd_Cmd(_LCD_CURSOR_OFF);
     Lcd_Out(1,1,txt1);
     Lcd_Out(2,1,txt2);
}
```

プログラムを記述した後はチュートリアル①と同様の手順でコンパイルをして、HEXファイルを生成後、書き込みを実行します。ボードの設定は特に注意する点はありませんが、ディップスイッチSW1はすべてOFFにしておいたほうがよいでしょう。

LCDの1行目に"HELLO"と表示され、2行目に"WORLD"と表示されれば成功です。

■プログラムの解説

sbit LCD_RS at LATB2_bit; ～
sbit LCD_D7_Direction at TRISB7_bit;
main関数の前にsbitタイプを用いてLCDの物理的な配線状態を定義します。
sbitタイプは、PICマイコンのSFR(レジスター)の特定のビットに対してアクセスするため機能を提供します。
ここでは、その1つの使い方としてLCDの接続状態をあらかじめ定義しています。sbitは、TRISレジスタに対しても特別な書式として記述することができます。LCD関連のライブラリを使用する場合には、main関数の前で、この定義を記述してください。
この配線記述は、dsPICF-900のボード配線にあわせて記述しています。

※この定義をmain関数の中で記述してしまうとエラーになります。

char txt1[] = "HELLO";
char型配列を宣言しています。char型は文字を扱う8ビットの型です。C言語では文字はASCIIコードにて扱われます。
通常配列を宣言する際には要素数(インデックス値)を[ ]内に入力しますが、[ ]内を空欄にすると自動的に要素数が設定されます。
配列や変数を宣言する際に初期値を代入できますが、char型に限り文字列で初期化できます。"(ダブルクォーテーション)で囲むと文字列として認識されます。

ADPCFG = 0xFFFF;
ADPCFGレジスタは、16個あるADコンバータ入力ピンのうち、どのピンをデジタルI/Oとして使うか、アナログ電圧入力として使うかを設定するレジスタです。アナログ入力にするピンに対応するビットは0にします。ここではすべてのピンをデジタルI/Oとして使用しますので、0xFFFFを代入しています。

Lcd_Init() ;
LCDを初期化する関数です。LCDを使用する前に記述します。

Lcd_Cmd(_LCD_CLEAR);
Lcd_Cmd( )関数は、LCDの制御コマンドを送信する関数です。
ここでは、画面をクリアーするコマンドと、カーソル(描画位置に点滅するカーソル)を表示しない設定にしています。
※mikroCのマニュアルは、"Help"→"Help"で見ることができます。

Lcd_Out(1,1,txt1) ;
Lcd_Outは4ビットモードでLCDに文字や文字列を表示する時に使用します。書式は下記の通りです。

Lcd_(行, 列, char *text);

*textは、char型のポインタという意味です。ここでは、配列名を記述することで配列の0番目の要素のアドレス値を指定しています。

*textは、char型のポインタという意味です。ここでは、配列名を記述することで配列の0番目の要素のアドレス値を指定しています。
なお、ポインタを使用して次のページのようにプログラミングしても同様の動作をします。ポインタについての詳細はC言語の専門書籍を参照ください。

チュートリアル③ ～非同期式シリアル通信とADコンバーター～

■プログラムの作成
非同期式シリアル通信(UART)を使用して、RS232C通信をするプログラムを作ります。RS232Cは、A系統(RF3、RF2)を使用します。

RS232C経由で文字'A'(0x41)を受信すると、ADコンバーターのチャンネル8(AN8)の値を8ビットの値で出力します。同様にして、文字'B'(0x42)を受信すると、チャンネル12(AN12)の値を8ビットの値で出力します。
なおUARTの通信速度は9600bps、データ長8ビット、1ストップビット、ノンパリティの設定とします。

```
void main() {

  char uart_rd;
  unsigned char AD_A, AD_B;

  TRISB = 0xFFFF;
  UART1_Init(9600);
  Delay_ms(100);

  while (1) {
    AD_A = ADC1_Read(8);
    AD_B = ADC1_Read(12);

    if (UART1_Data_Ready()) {
      uart_rd = UART1_Read();
      switch(uart_rd){
       case 'A': UART1_Write(AD_A);break;
       case 'B': UART1_Write(AD_B);break;
      }
   }
  }
}
```

■dsPICF-900ボードの設定
このプログラムを正しく動作させるため、下記のようにdsPICF-900ボードの各種設定をセットしてください。ボードの設定が正しくないとプログラムは動作しませんので、設定内容とプログラムを比べて確認してください。

・SW1及びSW2はすべてOFF
・SW3及びSW5はすべてOFF
・SW4はRS232AのRF3とRF2のスイッチをON、それ以外はOFF
・LCDはPORTBと接続されているため取り外してください。
・J2はRB8をジャンパーソケットでショート
・J3はRB12をジャンパーソケットでショート

■コンパイルと書き込みの実行

これまでのチュートリアルと同様、ソースプログラムをビルドして、HEXファイルをdsPICF-900のマイコンに書き込んでください。

■パソコン側の準備

このプログラムでは、RS232C通信を行います。パソコンのRS232CポートとdsPICF-900の"RS232A"とかかれたポート(左側のコネクタ)を全結線ストレートケーブルで接続してください。

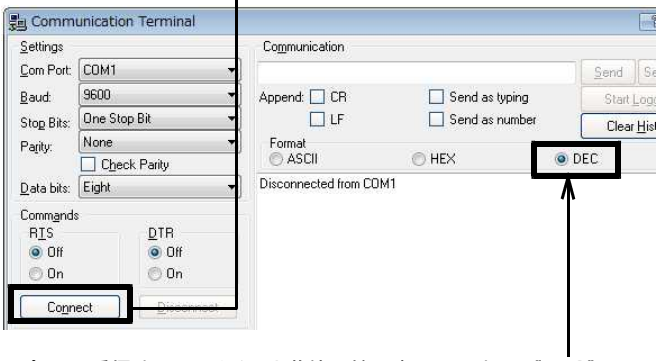
RS232Cのターミナルソフトは mikroCに付属しています。このターミナルソフトを使用すると大変便利に通信が行えます。
mikroCのツールバーにある"USART Terminal"のアイコンをクリックします。

ターミナルソフトが起動します。
"Settings"グループにて通信に関する基本的な設定を行います。

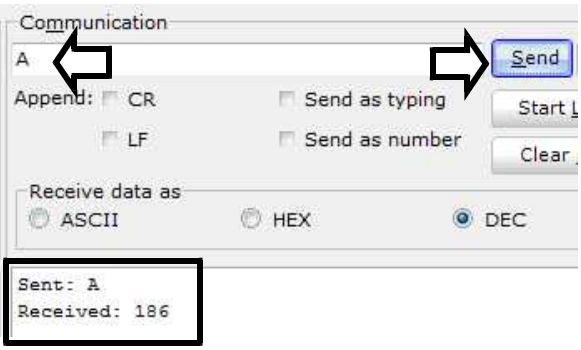
今回は、通信速度9600bps、1ストップビット、ノンパリティーの設定にしますので、上記のように設定してください。
"Com Port"は、dsPICF-900と接続したRS232Cのポート番号を設定してください。その他の設定はデフォルト設定にしておきます。

■動作確認

ターミナルソフトの"Connect"ボタンを押すとRS232C通信が開始されます。



今回は受信するデータを10進数の値で表示しますので"DEC"にチェックを入れます。
ターミナルソフトの"Communication"のテキストボックスに大文字でAと入力して"Send"ボタンをクリックします。
受信ウインドウに Sent:A と表示され"Received: xxx"と数値が表示されれば成功です。



"Received"で返る値は、ADコンバーターの値ですので、値は不定です。'A'を送信した場合には、AN8の値ですので、可変抵抗器P3で設定した値が返ります。

試しにP3ボリュームを回転させて、'A'を送信してみましょう。回転させる毎に異なった値が返ることが確認できます。ボリュームを左に回しきった状態の時は 0 が返ります。
同様にして、'B'も試してみましょう。'B'を送信すると、AN12のADコンバーターの値が返ります。P4ボリュームを回すと変える値が変わることが確認できます。
なお、いずれもADコンバーターの分解能は10ビットですが、シリアル通信側は8ビット長なので、最大値は255となり、255を過ぎると、0に戻ります。ボリュームのちょうど中間部分が0になります。
タクトスイッチのRE0をクリックすると数値"0"が、RE1では"1"が、RE2では"2"がdsPICF-900から送信されます。
スイッチを押している間数値が出力されるため、一度スイッチを押してもたくさんの数字が表示されます。

■プログラムの解説

TRISB = 0xFFFF;

PORTBは全ビット入力ピンに設定しています。今回は、AN8とAN12だけ入力ならばよいので、TRISB=0x1100 としてもよいです。

Uart1_Init(9600);

UART1の初期化を行う関数です。( )内には通信速度をbps単位で記述します。ここでは9600bpsに設定しています。

delay_ms(100);

UARTのユニットが初期化するのに約100ミリ秒かかりますので、100ミリ秒の遅延を挿入しています。delay_ms( )関数は、引数で指定した秒数(単位はミリ秒)だけ遅延を作ります。

AD_A = ADC1_Read(8);

Adc1_Read()関数は、( )内に指定したチャンネルからAD変換の値を取得して返します。本来PIC24FシリーズのADコンバーターは分解能が10ビット長ですが、ここでは、この値を後にUARTで送信する関係上、符号無しの8ビット長変数(char型)に代入しています。
ここでは8を指定してるため、AN8(RB8)に印加された電圧値の値を取得します。同様にして、次の行でAN12(RB12)からも値を取得します。

if (Uart1_Data_Ready()) {

Uart1_Data_Ready()関数はUARTの受信バッファにデータが格納されると真になります。これをif文で検出することでデータの受信を監視します。

uart_rd = Uart1_Read()

char型の変数uart_rdに受信データを取り込みます。1バイトサイズのデータのみ受信できます。連続するデータの場合には、配列とポインタを使用します。

switch (uart_rd){

switch文は、( )内の評価式を評価してcaseで指定された値のルーチンに処理を分岐します。評価式にuart_rd変数を指定し、UARTで受信したデータの内容によって処理を分岐しています。

caseの後には値を記述し、評価式がその値になると、そこにジャンプします。ここでは、'A'及び'B'を指定することで、受信した文字によって処理を分岐しています。

C言語では、1文字の場合には ' (シングルクォーテーション)で囲います。(文字列はダブルクォーテーションです。)

なお、break文を記述すると、switch文から処理が抜け出します。

Uart1_Write(AD_A);

Uart1_Write()関数は、UART経由で( )内のデータを出力します。( )内は符号なしのshort型又はchar型変数が指定できます。

ここでは、ADC1_Read()関数で取得した値をUARTで出力しています。

チュートリアル④ ～TMR1を使用した割込プログラム～

■TMR1とは・・

PIC24シリーズのデバイスには、TMR1という内蔵タイマーが搭載されています。TMR1は、16ビットのカウンターで、内部クロックか外部クロックに同期させてカウントを行わせることができます。

TMR1は、PR1レジスタの値と常に比較されるようになっており、TMR1の値と、PR1レジスタの値が一致した時、割込が発生する仕組みになっています。また割込発生後は、TMR1の値は0に初期化されます。

■PIC24シリーズの割込について

PIC24シリーズでは、割込の種類(何のイベントで発生した割込か)によって、ジャンプするベクタ(ジャンプベクタと呼びます)が異なるベクタ方式という割込を採用しています。8ビットのPICマイコンに比べ格段に構造が複雑になっています。これにより、様々な割込に応じて異なる割込ルーチンを作成でき、より複雑な割込プログラムを作ることができます。

ジャンプベクタは、プログラムメモリーのアドレスに割り当てられています。割込が発生すると、その割込の種類によって、ジャンプベクタが変わります。どんな割込が発生した時、どのジャンプベクタへジャンプするのかは決められており、例えば今回使用するTMR1割込の場合には、ベクタ番号は11となります。このように割込の種類と、飛び先アドレスの一覧を主ベクタテーブル(IVT)と呼びます。IVTは、プログラムメモリーの0x00014～0x000FEに配置されています。
(割込ベクターテーブルの一覧は、PIC24シリーズのデータシートに記載がありますので、併せてご覧ください。)

IVTの他にもう1つベクタテーブルが用意されており、これを副ベクタテーブル(AIVT)と呼びます。これは、プログラムメモリーの0x00114～0x001FEに配置されています。これは、IVTが何らかの事情で使用できない場合(※1)に、IVTに変わってAIVTを使用することができることを意味します。基本的には、IVTを使用すると覚えておいてください。

※1:PIC24では、プログラムメモリーにデータを書き込む場合、96バイト単位で書き込まなければならないという決まりがあります。
ITVに書き込みを行おうとすると、同時にリセットベクタも書き込みを行う必要が出てきます。例えばブートローダーを書き込んでいる場合などでは、リセットベクタ(0x0002番地)とIVTが隣り合っているため、IVTを書き換えるには、リセットベクタも同時に書き換えを行わなくてはならなくなり、ブートローダーで不具合が発生した場合、リセットベクタが不正になってしまうことがあるため、IVTにはさわらずに、割込を使用する場合には代わりにAIVTを使用することがあります。

また、16ビットPICでは、割込に順位が付くようになりました。これまでの8ビットのPICでは、割込は同時には1種類だけしか選択できませんでしたから、発生した割込の順位などは最初から必要ありませんでした。しかしdsPICでは複数の割込をベクタ方式で処理できるため、割込に優先順位を持たせてあります。

優先順位は、レベル0～レベル15までの16段階です。15が最も優先順位が高い割込になります。TMR1割込のような汎用割込の場合には、レベル7～レベル0が設定されます。

では順位はどのように決まっているかというと、IVTの並び順となっています。IVTの値が小さい方が、レベルが高い、すなわち優先的に割込として受け付けられるということです。

例えば、TMR1の割込はIVTでベクタ番号3です。対してADコンバーターの読み取り完了時に発生する割込はベクタ番号13です。よって、2つの割込が同時に発生した場合には、TMR1の方が優先されるということになります。

さらにCPUには割込許可レベルというものがあり、これも0～15で設定します。これは、CORCONレジスタのビット3と、SRレジスタのビット5～7の計4ビット(CORCONレジスタのビット3が最上位ビット)で指定するもので、ここで設定した値よりも、優先順位が高い割込が、実際の割込として許可される仕組みになっています。

■プログラムの内容

TMR1割込を使用して、250ミリ秒毎に割込を発生させて、PORTBの出力を、250ミリ秒ごとに反転させるプログラムを作ります。

■プログラムの仕組み

今回は、dsPICF-900のMPUボードに搭載されている8MHzの水晶発振子をクロックとして使用します。クロックは逓倍せずにそのままCPUのクロックとして8MHzとして使用します。

TMR1のクロック源は内部クロックとします。PIC内部では動作クロックが1/4になります。よって、内部は2MHzで動作しています。
時間に直すと1周期500nsです。タイマーの時間は下記の式で求められます。

TMR1時間=(1/4×動作クロック) × プリスケーラー分周比 ×
(PR1レジスタの設定値-1)

今回は下記の式により約250ミリ秒を作ります。

500ns×256×(1954-1) = 249.984ミリ秒

きれいに割り切れないため、若干の誤差が生じますがこれは、クロック周波数を見直すことできれいに割り切れる値になります。
今回は、8MHzのクロックを使用している関係上、若干の誤差が生じる計算になります。

■プログラムの作成

新しいプロジェクトを作成して、プログラムを作ってください。プロジェクトの作成方法等は従来の方法と全く違いはありません。

```
void Timer1Int() iv IVT_ADDR_T1INTERRUPT{
  T1IF_bit = 0;
  LATB = ~ PORTB;
}

void main() {
  ADPCFG = 0xFFFF;
  TRISB = 0;
  LATB = 0;

  T1IF_bit = 0;
  T1IE_bit = 1;
  T1CON = 0x8030;
  IPC0 = 0x7000;
  PR1 = 1954;
}
```

■dsPICF-900ボードの設定

SW1の"PORTB_L"と、"PORTB_H"のスイッチだけをONにしてその他はすべてOFFに設定します。SW3もすべてOFFにします。

■プログラムの解説

void Timer1Int() iv IVT_ADDR_T1INTERRUPT{

TMR1の割込が発生した時に実行されるルーチンです。
mikroCでは、ivキーワードがあり、ivキーワードに続いて、割り込みの種類を指定することで、その割り込みが発生した場合に、{ }内に記述されたルーチンが実行されます。

通常は、main関数が実行されていますが、割り込みが発生した時にこの関数が実行されます。

T1IF_bit = 0;

TMR1割込発生フラグを初期化しています。IFS0レジスタのビット3に0を代入しています。割込先ルーチンでは、必ず対応する割込発生フラグを初期化しないと、次の割込を発生させられませんのでここで初期化しています。
mikroCでは、レジスタの特定のビットを指定する場合には、"ビット名_bit"で指定することができ、代入演算子で値を代入できます。

LATB =~ PORTB;

ポートの状態を反転させています。LATBレジスタは、現在のポートの状態ではなく、PORTBレジスタの値をそのまま反映しています。よって、PORTBの状態を毎回反転させることで点滅させています。

T1IE_bit = 1;

IEC0レジスタのビット3にある、TMR1割込の許可制御ビットを1にしてTMR1の割込を有効にしています。

T1CON = 0x8030;

T1CONレジスタは、TMR1の各種設定をするためのレジスタです。
ここでは、「外部ゲートOFF・プリスケーラー分周比256・クロック源は内部クロック」という設定にしています。

IPC0 = 0x7000;

IPC0レジスタでは、割込の優先順位を制御します。
ビット14～11がTMR1の割込レベルの設定ビットになっており、000で禁止、111でレベル7(最高)となります。
今回は、TMR1割込しか使いませんので、最高レベルの111に設定しています。

PR1 = 1954;

PR1レジスタの値です。TMR1は常にここの値と比較され、ここの値と一致した時に割込を発生させます。
今回は前のページに記載した計算によって求めた値を代入しています。

※今回のプログラムでは、main関数内ではレジスタの設定をしているだけです。main関数内に別の処理を記述していくと、割込が発生するごとに割込ルーチンが実行され、割込が完了すると、main関数が再度実行され、これを永久に繰り返すことになります。

チュートリアル⑤　～ICD機能を使ったデバッグを体験する～

■ICD機能とは?

ICDとはインサーキットデバッグの略で、ターゲットボード上のデバイスに実際にプログラムを書き込み、動作させながらパソコン上でデバッグを行う手法のことです。

例えば現在実行している行をハイライト表示させながら実機での動作が見られますので、プログラムの挙動を簡単に把握できます。また、1行ずつプログラムを実行したり、変数やレジスタの現在の値をリアルタイムに閲覧することができたりします。

本セットに付属のmikroCには、C言語レベルでデバッグのできるICD機能が搭載されています。dsPICF-900と組み合わせることで、ICD機能を簡単に体験することができます。

■ICD機能を使うには

ICD機能を使用する場合、プログラムには特に大きな変更点はありませんが、delay_(ms)などのdelay関数が入ったプログラムは正しく動作しません。ICDを使用する場合には、delay関数をプログラム中からなくして実行してください。また、関数によっては、1つの関数で大量の処理を実行するため、ステップ実行(1行ずつ実行)では、時間がかかりすぎる場合があります。

ICD機能を使用する場合、コンパイル時に"Project Settings"から、"Build type"を"ICD debug"設定にする必要があります。このチュートリアルを通してICDの使い方を一通り体験してみましょう。

■プログラムの内容

LCDに文字"microtechnica"を表示されます。但しICD機能の動作を確認するため1文字ずつ表示するプログラムを作ります。

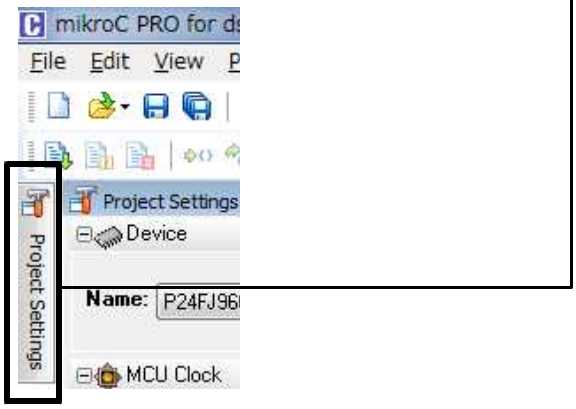
■プログラムの作成

新しいプロジェクトを作成して、プログラムを作ってください。
プロジェクトの作成方法等は従来の方法と全く違いはありません。

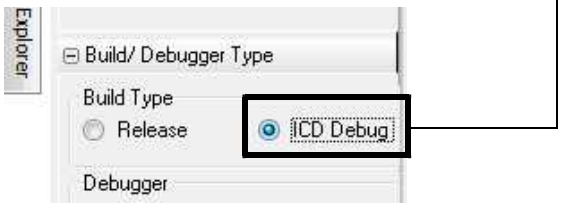
```
sbit LCD_RS at LATB2_bit;
sbit LCD_EN at LATB3_bit;
sbit LCD_D4 at LATB4_bit;
sbit LCD_D5 at LATB5_bit;
sbit LCD_D6 at LATB6_bit;
sbit LCD_D7 at LATB7_bit;

sbit LCD_RS_Direction at TRISB2_bit;
sbit LCD_EN_Direction at TRISB3_bit;
sbit LCD_D4_Direction at TRISB4_bit;
sbit LCD_D5_Direction at TRISB5_bit;
sbit LCD_D6_Direction at TRISB6_bit;
sbit LCD_D7_Direction at TRISB7_bit;

void main(){

   char text[14]="microtechnica";
   int i;

   AD1PCFG = 0xFFFF;
   Lcd_Init();
   Lcd_Cmd(_LCD_CLEAR);
   Lcd_Cmd(_LCD_CURSOR_OFF);

   for(i=0 ; i<13 ; i++){
      Lcd_Chr_Cp(text[i]);
   }
}
```

■コンパイルと書き込み

プログラムが書き終わったら、コンパイルをしますが、この時従来とは違い、Build typeをICD用に変更します。画面左下の"Project Settings"タブをクリックします。ウインドウが展開します。



ウインドウが表示されたら、その中の"Build/Debugger Type"の部分にある"Build Type"のチェックボックスのうち、"ICD debug"にチェックを入れます。



チェックを入れたら、従来通り、メニューバーのビルドして、コンパイルが終了したら従来どおり、PICマイコンへ書き込みを行ってください。この作業はこれまでと同じです。

"ICD debug"オプションを指定してコンパイルしたプログラムは、PICマイコンへ書き込んだ後にすぐには動作しません。

■動作の前にボードの確認をしましょう

実際にプログラムを走らせる前には、必ずボードの設定状態を確認してください。ボードの設定が正しくないと動作しないことがあります。

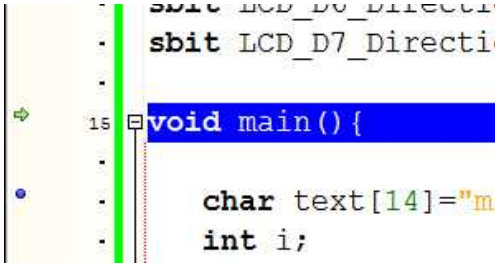
・SW1とSW2はすべてOFF側に設定してください
・LCDを取り付けてください

■ICDを動作させてみる

ICDを実行して、プログラムの動作内容を確認しましょう。
LCDに何も文字列が表示が出ていないことを確認します。文字列が表示されている場合には一度dsPICF-900の電源を切断してLCDの画面をクリアした状態で下記を実行してください。

1 メニューバーの"Run"→"Start Debugger"をクリックします。
キーボードの"F9"キーでも同様の操作ができます。

2 void main() { の行が青色バーでハイライトされます。
ICDでは実行される行が青色バーでハイライト表示されます。
また、行数表示の左隣に➡マークが表示され、実行している行が分かります。



※void main() { の行が青くハイライトされず、別のアセンブラの画面が表示されてしまった場合には、メニューバーの"Run"→"DisAssembly mode"をクリックしてください。

3 ICDでは、1行ずつ実行する場合、LCD関連のライブラリーは動作が複雑で時間がかかるため、ここでは一気にfor文の所までは通常通りプログラムを動作させてしまいましょう。

"for(i=0 ; i<13 ; i++){ " の行をクリックします。カーソルがこの行に移動していることを確認してください。

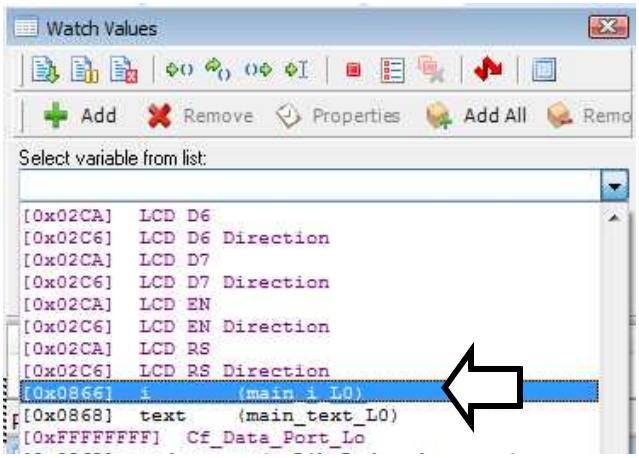
4 カーソル行までプログラムを一気に動かす方法でfor文位置までプログラムを移動させます。
メニューバーの"Run"→"Run to Cusor"をクリックします。キーボードのF4キーでも同様の操作ができます。
for文の一行が青色バーでハイライト表示されます。

5 LCDにはまだ文字は表示されていないはずです。
この状態で1行ずつ実行するステップ実行を行い、1文字ずつLCDに文字が表示されるのを確認しましょう。
キーボードのF8キーを押します。又はメニューバーの"Run"→"Step Over"をクリックします。F8キーを押すと、青色バーでハイライト表示された部分が移動します。
F8キーを2回押すと、LCDに"m"と表示されます。

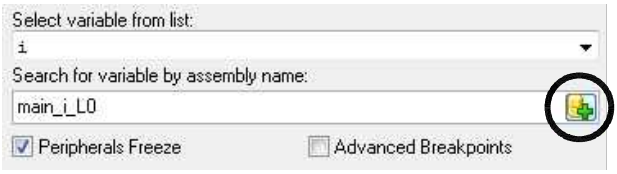
6 同様にして、F8キーを何回も押すと、プログラムはFor文を繰り返し実行して、1文字ずつLCDに文字が表示されていきます。5文字程度表示されたら次の手順を実行します。
なお、F8キーを早く押してしまうと、ICD機能の処理が間に合わずエラーが表示されてしまうことがあります。F8キーを押してステップ実行を進める場合には、必ず青い実行行を示すバーが次の行に移動してからF8キーを押すようにしてください。

7 ICDを実行すると、Watch Valueウインドウが画面の右側に表示されます。ウォッチウインドウは、変数やレジスタの値をリアルタイムで閲覧できるウインドウです。ここでは変数iの値が1ずつ増えていく様子を確認しましょう。

8 変数iを追加します。
"Select Variable form list"のプルダウンをクリックして、その中から "[0xXXXX] i (main_i_L0)" と表示されている部分クリックします。※XXXXは値が入ります。



9 Addボタンを押して、追加します。



10 リストに追加されるとNameとValue、Addressが表示されます。
Valueの値は現在の変数iに格納されている数値です。
F8キーを押してFor文を実行すると、1ずつiの値が増えていく様子を見ることができます。

値が更新されると数値は赤文字で表示されます。

■ICD機能を使用しない場合の注意
ICD機能を使用する場合、コンパイル時のBuild typeをICD用に設定しました。このICD用に設定されて生成されたHEXファイルは、通常動作はしません。ICD機能を使用しない通常のHEXファイルをビルドしたい場合には必ずビルドする前に"Project Settings"の"Build type"から"Release"にチェックを入れて、ビルドを行ってください。

サンプルプログラムについて

mikroCのサンプルプログラムが、mikroCのインストールされているディレクトリ内に収録されています。通常下記のディレクトリに収録されています。

C:¥Users¥Public¥Documents¥Mikroelektronika ¥mikroC PRO for dsPIC¥Examples ¥Development Systems¥LV24-33 v6